COURSE DATA

Data Subject

Code	36411
Name	Programming fundamentals
Cycle	Grade
ECTS Credits	6.0
Academic year	2022 - 2023

Study (s)

Degree	Center	Acad. year	Period
1406 - Degree in Data Science	School of Engineering	1	First term

Subject-matter

Degree	Subject-matter	Character
1406 - Degree in Data Science	3 - Informatics	Basic Training

Coordination

Name	Department
FERRIS CASTELL, RICARDO	240 - Computer Science

SUMMARY

In this subject it is about learning the basic knowledge of what a computer is, what are its basic components, potential uses and limitations.

It will be a question of getting a sufficient knowledge of the design of algorithms through structured programming, as well as of the fundamental data structures, which progressively allows more complex problems to be tackled later, both from an analytical and numerical point of view.

With regard to the practical part, in this subject we will try that students consolidate the knowledge seen in the theoretical part both regarding the knowledge of the computer and the basic tools for its use and that they acquire abilities of development of programs in a structured programming language of general purpose and extended use in the field of Data Science.

Theory lessons will be taught in Spanish and practical and laboratory lessons as according to the information sheet available on the web page of the degree.

PREVIOUS KNOWLEDGE

Relationship to other subjects of the same degree

There are no specified enrollment restrictions with other subjects of the curriculum.

Other requirements

None.

COMPETENCES (RD 1393/2007) // LEARNING OUTCOMES (RD 822/2021)

1406 - Degree in Data Science

- (CG01) Knowledge of basic subjects and technologies that enable students to learn new methods and technologies, and to provide them with versatility to adapt to new situations.
- (CG06) Ability to access and manage information in different formats for subsequent analysis in order to obtain knowledge from data.
- (CT02) To be able to complete technical, scientific, social and human training in general, and to organise self-learning with a high degree of autonomy.
- (CT05) Ability to evaluate the advantages and disadvantages of different methodological and / or technological alternatives in different fields of application.
- (CE02) To methodologically know and apply the programming techniques and the algorithms necessary for the efficient processing of information and the computer resolution of problems that use large volumes of data.
- (CE11) Ability to design and implement data acquisition, its integration, transformation, selection, verification of its quality and veracity from different sources, taking into account its character, heterogeneity and variability.
- (CB1) Students must have acquired knowledge and understanding in a specific field of study, on the basis of general secondary education and at a level that includes mainly knowledge drawn from advanced textbooks, but also some cutting-edge knowledge in their field of study.
- (CB2) Students must be able to apply their knowledge to their work or vocation in a professional manner and have acquired the competences required for the preparation and defence of arguments and for problem solving in their field of study.

LEARNING OUTCOMES (RD 1393/2007) // NO CONTENT (RD 822/2021)

To know the basic characteristics and operation of the elements that make up a computer, detailing both the physical parts that make it up (central process unit, memory...) and the logical parts that make it work (operating system, programs, etc.) (CB1, CG01, CT01).

To know and use the different types of data, simple and structured, appropriately for the computerised representation of data (CE11).

To develop, maintain and adapt structured codes that are robust, efficient and safe (CT05). Apply the concept of algorithm and the structured programming for the resolution of problems (CB2): Analysis of the problem; Design of the algorithm (descending or modular design and refinement by steps); Programming of the algorithm (CT02, CE02).

To program algorithms in Python programming language. To know the specific details of programming seen in the theoretical module (data types, variables, etc.) and practice the different control structures and the use of functions to perform a modular treatment of the problems.

To know how to develop programs that work with different data entry formats and generate information in the desired output format (CG06, CE02, CE11).

DESCRIPTION OF CONTENTS

1. Introduction

Basic concepts.
Internal structure of the computer: control unit, arithmetic-logic unit, memory unit, input unit and output unit.
Languages and programming paradigms: procedural languages and declarative languages.
Operating system

2. Programming in high-level languages

Algorithms.
Characteristics of high-level programming languages: Objects and references, simple data types, strings and lists, and data input and data output.
Phases in the realization of a program: Analysis of the problem, design of the algorithm and programming of the algorithm.

3. Structured programming

Theorem of structured programming.
Design of structured programs.
Control structures: sequential structure, conditional structure and iterative structure.

4. Files

Basic concepts of files: Type of access, Logical files and physical files and Binary and text files.
Processing of files.

5. Modular programming

Module definition: Modular programming, Definition of subprograms: Functions, Parameters of a subprogram and Scope of identifiers.
Recursivity.

6. Structured data types

More about strings and lists.
Collections.
Introduction to Classes.

WORKLOAD

ACTIVITY	Hours	% To be attended
Theory classes	28,00	100
Laboratory practices	20,00	100
Classroom practices	12,00	100
Development of group work	10,00	0
Development of individual work	20,00	0
Preparation of evaluation activities	10,00	0
Preparing lectures	15,00	0
Preparation of practical classes and problem	35,00	0
TOTAL	150,00	

TEACHING METHODOLOGY

In the in classroom theoretical activities the topics of the subject will be developed, providing a global and integrating vision, analysing in greater detail the key aspects and of greater complexity, promoting, at all times, the participation of students (CB1). These activities are complemented by practical activities with the aim of applying the basic concepts and expanding them with the knowledge and experience that are acquired during the performance of the proposed works (CB2). They include the following types of classroom activities: types of problems and questions in the classroom; discussion and problem solving sessions and exercises previously worked by students; laboratory practices; individual evaluation questionnaires in the classroom with the presence of the teaching staff (CG01, CG06, CE02, CE11).

In addition to in classroom activities, students must perform personal tasks (outside the classroom) on: monographic works, directed bibliographic searches, questions and problems, as well as the preparation of classes and exams (study) (CT01, CT02). These tasks will be carried out mainly individually, in order to enhance self-learning, but will also include jobs that require the participation of small groups of students (4-6) to promote capacity for integration in work groups (CT03).

The e-learning platform (Virtual Classroom) of the University of Valencia will be used to support communication with the students. Through it you will have access to the didactic material used in class, as well as the problems and exercises to solve.

EVALUATION

The evaluation of the subject will be carried out by means of:

- Continuous evaluation, based on participation and degree of involvement in the teaching-learning process, taking into account the regular attendance at the planned activities and the resolution of proposed questions and problems. Occasionally, oral presentations can take place (individually and/or in groups) to assess the ability to produce documents and transmit knowledge (N_Continuous). In classroom activities are non-recoverable (SE3).
- Individual objective test, consisting of several controls throughout the semester, and a final exam, which will include both theoretical-practical questions and problems (N_Exams) (SE1).

N_Exams = 60% Controls + 40% Final Exam

The value of all controls will be the same.

Controls are not recoverable.

- Evaluation of practical activities from the achievement of objectives in the laboratory sessions and problems, and the preparation of papers/reports, including the final project (N_Practices) (CB1, CB2, CG1, CG6, CT1, CT3, CT5, CE2, CE11). Attendance to laboratory lessons is compulsory to pass the subject in first call (SE2).

N_Practices = 30% Practices work + 70% Final project

At least a 4 has to be obtained in the final project to make the average.

The final score of the subject will be:

Final score = 20% N_Continuous + 50% N_Exams + 30% N_Practices

It will be necessary to obtain at least 3.5 out of 10 in each part to be able to make the average.

In second call it is possible to improve the final practice (if you have not attended practical lessons, the final practice will have to be defended before the teacher and will be 100% of the practice note), improve the note of the questions and problems proposed throughout the semester to be done at home and improve the final exam (the weight of controls will be reduced to 20% in N_Exams). The weight of each section will be the same as in the first call, as well as the conditions to pass the subject.

In any case, the assessment of the subject will be done in accordance with the assessment and qualification Regulation of the University of Valencia for degrees and masters

https://webges.uv.es/uvTaeWeb/MuestraInformacionEdictoPublicoFrontAction.do?accion=inicio&idEdictoSeleccionado=5639

REFERENCES

Basic

- [Kent D. Lee (2014)] Python Programming Fundamentals (Spinger). https://link.springer.com/book/10.1007/978-1-4471-6642-9
- [A. Marzal, I. Gracia, P. García (1993)] Introducción a la programación con Python 3. https://www
r9fC1vXiAhUUXRUIHerpDWgQFjAAegQIAhAC&url=http%3A%2F%2Frepositori.uji.es%2Fxmlui%2Fbitstream%

Additional

- [A. Downey, J. Elkner, C. Meyers (2002)] Aprenda a Pensar Como un Programador con Python (Green Tea Press). Traducido por M.A. Vilella, A. Arnal, I. Juanes, L. Amurrio, E. Andia, C. Ballardini. https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf