Programming techniques

Academic Year: (2022 / 2023) **Review date: 09/05/2023 16:30:57**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: RUIZ MEZCUA, MARIA BELEN

Type: Compulsory ECTS Credits : 6.0

Year : 2 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

None

OBJECTIVES

In a generic way, the subject aims for the student to acquire a series of knowledge, skills and abilities in relation to the basic elements of programming. Data structure, syntax, pseudocode.
Additionally students will acquire skills to propose solutions to solve problems in teams.

DESCRIPTION OF CONTENTS: PROGRAMME

1. Introduction- Programming languages- Compilation and execution of programs

2. Programming foundations- Programming paradigms: structured programming, modular programming.- Elements of a program: data and algorithms-Basic programming tools: algorithms, flow diagrams and pseudocode.

3. Programming in Python - Characteristics of the Python language Data types. Input and output .

4. Flow Control - Conditional Indentation - Loops: while loop and loop over lists

5. Functions: Built-in functions (BIF) and custom developed functions. Documentation and test of functions.

6. Python sequences. Mutable (lists, sets) and immutable sequences (strings, tuples). Sequence tour. 'Slicing'. More common operations.

7. Basic techniques of identification and detection of errors, tests. Testing and debugging.

8. Introduction to access to external resources. Basic file management modules. CSV files. OS modules. Modules, basic connection operations with relational databases (SQLite ...)

9. Introduction to external resources. Libraries. Pandas, MatPlotLib. Communities.

LEARNING ACTIVITIES AND METHODOLOGY

Acquisition of theoretical and practical knowledge (3 ECTS) with theoretical classes, tutorials, to forums and chats of the global classroom platform, seminars and workshops on tools, problem solving the case studies, both individually and in groups

Acquisition of competences and skills (3 ECTS) in solving problems, proposing a solution, writing the pseudocode, programming in python and presenting the solution to the class..

The days and hours of the tutorials can be viewed in the space dedicated to the subject in Aula Global
It is very important that the students keep the subject up to date and try to solve on their own the exercises and practical cases proposed by the teacher, prior to their resolution in the classroom. Success in the subject depends to a large extent on the constancy of the student's personal work.

ASSESSMENT SYSTEM

% end-of-term-examination/test:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

- Given that this subject has a great practical component, the acquisition of cognitive and attitudinal skills is of great importance. The evaluation system is based mainly on the measurement of these competences. These competences will be measured through the following instruments:
Continuos evaluation
¿ Final exam: 40%. It consists of the following elements:
o Written (can be online depending on the teaching situation) - 15%
o Computer exercises resolution: 15%
o Defense of practice: 10%.
¿ Continuous evaluation: 60%, distributed in turn into:

o Delivery of exercises. 25%
o First part of team practice 10%
o Second practical delivery in team 25%

The continuous assessment test will be individual, even if you opt for the option of defending team practice, the answer to the questions raised by the teaching staff will be individual.

To pass the evaluation, it will be necessary to achieve at least 30% of the grade in each of the two blocks.

Extraordinary call:
Students who do not pass the subject in the ordinary call may choose one of the following options in the extraordinary call:

1. Take the individual exam again and keep all the marks obtained in the continuous evaluation. This exam will score 50% of the grade

2. Take a final test, renouncing the grades obtained in the continuous evaluation.The final test will include a written part and solving exercises, and will include the entire grade.

BASIC BIBLIOGRAPHY

- González Duque, Raúl Python para todos , (Distribuido con licencia Creative Commons. Disponible en http://mundogeek.net/tutorial-python/).

- Peña, Rosalía Resolución de problemas para ingenieros con Python® estructurado, ibergaceta, 2016

- Severance, Charles. Python for Everybody: exploring data in Python 3., Distribuido bajo licencia Creative Commons by-nc-sa..

BASIC ELECTRONIC RESOURCES

- . The Python Wiki: http:// //wiki.python.org/moin/FrontPage

- . Documentación oficial de Python:: http://docs.python.org/3.6/

- . The Python Wik: http://wiki.python.org/moin/FrontPage

- . Pythontutor (simulador en línea de código Python): http://www.pythontutor.com

- . ¿ PSInt (para la confección de diagramas de flujo y pseudocódigo).: http://pseint.sourceforge.net/

- . ¿ IDLE (entorno de desarrollo básico para Python).: http://www.python.org/downloads/

- . ¿ Spyder (entorno de desarrollo para Python más potente y versátil que el anterior): http://www. anaconda.com/distribution/

- . The Python Wiki: http:// //wiki.python.org/moin/FrontPage

- Anaconda . ¿ Jupyter note books (Anaconda). : http://jupyter.org/

- Bartolomé Sintes Marco . Esta página forma parte del curso Introducción a la programación con Python : http:// //www.mclibre.org/consultar/python/

- Bartolomé Sintes Marco . curso Introducción a la programación con Python : http://www.mclibre.org/consultar/python/

- Spyder . ¿ Spyder (entorno de desarrollo para Python: http:// www.anaconda.com/distribution/