Course guide
230328 - AP - Learning with Python

Last modified: 25/05/2023

Unit in charge: Barcelona School of Telecommunications Engineering
Teaching unit: 701 - DAC - Department of Computer Architecture.

Degree: BACHELOR'S DEGREE IN TELECOMMUNICATIONS TECHNOLOGIES AND SERVICES ENGINEERING (Syllabus 2015). (Optional subject).

Academic year: 2023 **ECTS Credits:** 2.0 **Languages:** Catalan

LECTURER

Coordinating lecturer: Consultar aquí / See here: https://telecos.upc.edu/ca/estudis/curs-actual/professorat-responsables-coordinadors/responsables-assignatura

Others: Consultar aquí / See here: https://telecos.upc.edu/ca/estudis/curs-actual/professorat-responsables-coordinadors/professorat-assignat-idioma

PRIOR SKILLS

Students should have taken a first course on Object Oriented Programming

TEACHING METHODOLOGY

First half: Problem based learning
Second half: Project development

LEARNING OBJECTIVES OF THE SUBJECT

At the end of the course, students should:
1) Be familiar with the programming framework Spyder.
2) Given a set of language constructions and primitives, students should be able to solve particular programming problems.
3) Have improved their algorithmic programming skills.
4) Be familiar with scientific programming in Python with NumPy (Matlab style).
5) Be familiar with the graphic primitives of Matplotlif (Matlab-style graphics).
6) Students will have build a guided project of medium size in Python.
7) Students will have participated in a project that shows the MVC application architecture (Model ? Vision ? Control).

STUDY LOAD

Type	Hours	Percentage
Self study	30,0	60.00
Hours small group	20,0	40.00

Total learning time: 50 h

CONTENTS

Contents

Description:
Python
- The Python Interpreter (IPython). Python, a non-typed programming language.
- Data and variables. Objects and references (everything is an object in Python). Mutable and immutable data.
- Elementary sentences: multiple assignment, flow control, etc.
- Function and object calls. Passing parameters by reference.
- Data structures: strings, tuples, lists, maps, sets. Sequences.
- List comprehensions
- Modules in Python: functions and classes.
- Inheritance. In Python, everything is polymorphic.
- Exceptions

NumPy
- Arrays i matricial calculus. Some basic primitives.

Matplotlib
- Plot, scatter plot, ticks, labels, etc.

Related activities:
Guided working sessions at the lab. An example: a particular programming problem is formulated by the teacher, somehow related to the project, together with a few language constructs. Students should solve the problem with this constructs, and then compare their solution with an equivalent solution based on NumPy.

Full-or-part-time: 20h
Practical classes: 20h

GRADING SYSTEM

By attendance (students with more than two unjustified absences will be marked NP). Other than that, course marks will be determined by self-assessment, with the consensus from the teacher, based on specific tasks.

RESOURCES

Hyperlink:
- Nom recurs. Resource

Other resources:
The Python Tutorial: https://docs.python.org/2/tutorial/index.html />Tutorial interactiu: https://www.learnpython.org/ NumPy quickstart tutorial: https://docs.scipy.org/doc/numpy-dev/user/quickstart.html />Matplotlib tutorial: https://matplotlib.org/users/pyplot_tutorial.html />