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 Matplotlib (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).

Teaching methodology
Using an engineering project as a leading thread, we will work on guided exercises at the lab.

Learning objectives of the subject

Prior skills
Preferably, students should have taken and passed POO.

Teaching staff
Coordinator: Beatriz Otero
Others: Pau Bofill
Beatriz Otero

Study load

<table>
<thead>
<tr>
<th>Total learning time: 50h</th>
<th>Hours small group:</th>
<th>20h</th>
<th>40.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self study:</td>
<td>30h</td>
<td>60.00%</td>
<td></td>
</tr>
</tbody>
</table>
Content

<table>
<thead>
<tr>
<th>Contents</th>
<th>Learning time: 20h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Practical classes: 20h</td>
</tr>
</tbody>
</table>

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.

Qualification 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.

Bibliography