Degree competences to which the subject contributes

Specific:
1. Programming tools for solving problems related to engineering and bioprocesses.
2. Biological models and determination of their main properties.

Generic:
3. Ability to solve problems. LEVEL 3

Teaching methodology

In the class sessions, lectures will mainly be employed with participatory approach and problem solving. First there will be a brief presentation of the structure of the topic to facilitate organized information appropriate to the objectives specified. These sessions also incorporate room for student participation and their involvement through activities of short duration and resolutions of exercises in the classroom. The problem solving is apply primarily to small groups in the computer labs, in order to have access to the appropriate software. The autonomous learning will mainly focus on actions aimed at solving problems and doing exercises, as well as the preparation and implementation of simple programs in different environments. There will be questionnaires for self-assessment and for evaluation of contents through the virtual campus. Regarding group work, students will carry out a final project in order to prepare, implement and analyse a simulator to solve a problem in the context of biosystems engineering and bioprocesses.

Learning objectives of the subject

The programming and problem solving course will follow general training objectives, building students’ skills in learning and promoting attitudes for assessment, suitability and usefulness of models, algorithms, computer procedures and diverse programs. Essentially, the course provides the students with fundamental knowledge of programming, some basic tools in the use of specific programs, and some computer skills as a help to tackle problems in the field of biosystems engineering.

Taking full advantage of the matter, the students should
- Identify the decisive events in the history of computing to become aware of the evolution of computers and programming up to the current situation.
- Know, understand and use the basic concepts and principles of programing, algorithms, structures and types of variables.
- Be able to design simple algorithms, know how to write the corresponding pseudocodes, and draw the adequate flowcharts.
- Know and understand the software development cycle from specification or statement of the problem, going through the intermediate steps (processing scheme, design of the algorithm, and writing the code) to achieve the execution, and
the debugging mechanisms of algorithms and programs.
- Acquire the basics of structured traditional imperative programming and identify the elements that characterize the object-oriented programming to compare them.
- Know the basic elements of the syntactic structure and semantics of a programming language (Basic, Fortran or other) to be able to translate simple algorithmic designs.
- Use a spreadsheet (Excel or other) and a mathematical software (Maple or some other) for the approach and treatment of problems in biosystems engineering, as well as for analytical or numerical resolution.
- Become familiar with NetLogo, a multi-agent programming language, in order to handle a set of simulators with appropriate criteria.
- Understand and modify programs already developed in this NetLogo framework, as well as create their own simulation programs for research in various biological systems.

### Study load

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Hours large group: 40h</th>
<th>26.67%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hours medium group: 0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Hours small group: 20h</td>
<td>13.33%</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Self study: 90h</td>
<td>60.00%</td>
</tr>
</tbody>
</table>
## INTRODUCTION TO PROGRAMMING AND PROBLEM SOLVING

**Learning time:** 50h  
Theory classes: 12h  
Laboratory classes: 8h  
Self study: 30h

**Description:**  
Introduction to computing.  
Introduction to programming languages.  
Introduction to different strategies for solving problems.

**Related activities:**  
Activity 1: Lectures  
Activity 2: Individual written exams  
Activity 3: Problem solving and computer labs  
Activity 4: Final project

## ALGORITHMS AND DEVELOPMENT OF PROGRAMS

**Learning time:** 50h  
Theory classes: 15h  
Laboratory classes: 5h  
Self study: 30h

**Description:**  
(Fundamental notions: algorithm, basic algorithmic structures, variable types, input / output, flow chart, pseudo code, search algorithms, algorithms of order.  
General outline of the problem formalization of the algorithm specification, design, coding and implementation.  
Compilation process and interpretation process, linking (using libraries), execution, and analysis or debug programs.

**Related activities:**  
Activity 1: Lectures  
Activity 2: Individual written exams  
Activity 3: Problem solving and computer labs  
Activity 4: Final project
SPECIFIC SOFTWARE TO TACKLE THE PROBLEM SOLVING

Description:
Spreadsheet (Excel or other), their complements (or options) and programming for solving problems. Mathematical software (Maple or other) with their libraries to address numerical, algebraic analysis of problems. The platform free access software Netlogo: analysis, modification and implementation of the computational models implemented, and the creation of new programs for research and problem solving that require the formulation of discrete models.

Problem solving that requires the use of probability, arrays, continuous functions, discrete functions, optimization, linear programming, difference equations, and ordinary differential equations among other options.

Contextualization of problems applied to biosystems engineering, using different computational environments to identify the strategy for the resolution and using the appropriate software for their execution.

Related activities:
Activity 1: Lectures
Activity 2: Individual written exams
Activity 3: Problem solving and computer labs
Activity 4: Final project

Learning time: 50h
- Theory classes: 13h
- Laboratory classes: 7h
- Self study: 30h
### Planning of activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
<th>Support materials</th>
<th>Support materials</th>
<th>Specific objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACTIVITY 2: INDIVIDUAL WRITTEN EXAMS</strong></td>
<td>Individual assessment by individual written exam in classroom or computer lab. There will be two mid-term tests during the semester and a final test at the end of the course which will include all the contents developed during the course. Correction by the teacher who will provide the corresponding solutions.</td>
<td>Exam sheets and calculator, and where appropriate, specific software and some documentation.</td>
<td>Exam sheets and calculator, and where appropriate, specific software and some documentation.</td>
<td>Students should be able to propose, implement and execute simple programs or algorithms for solving various problems in the field of biosystems engineering. They should also be able to use distinct software to solve different types of problems.</td>
</tr>
<tr>
<td><strong>ACTIVITY 3: RESOLUCIÓ D'EXERCICIS I PROBLEMES</strong></td>
<td>This activity is developed in sessions of two hours, or one hour, either individually or in groups. Before the activity in the computer room the students should read the documentation on the activity in order to familiarize themselves with the goals to be achieved.</td>
<td>Documentation of the activity available in Atenea and/or a printed copy, and specific software.</td>
<td>Documentation of the activity available in Atenea and specific software.</td>
<td></td>
</tr>
<tr>
<td><strong>ACTIVITY 4: FINAL PROJECT</strong></td>
<td>Preparation of a project to propose, design, implement and run a program to deal with a problem in the field of biosystems engineering, in which topics developed during the course can be applied, choosing the appropriate computing environment for the resolution of the different tasks involved.</td>
<td>Documentation of the activity available in Atenea and specific software.</td>
<td>Documentation of the activity available in Atenea and specific software.</td>
<td></td>
</tr>
</tbody>
</table>

#### Hours

- **ACTIVITY 2**: 2h
- **ACTIVITY 3**: 35h
- **ACTIVITY 4**: 15h

---

*Last update: 04-02-2020*
Description of the assignments due and their relation to the assessment:
Evaluation of the documentation generated.

Specific objectives:
At the end of this activity students should be able to cover the different stages to achieve a simulator that responds to a specific problem, organizing information regarding the problem, choosing the right software, designing and implementing different parts of the code, and analyzing the simulation results.

Qualification system

N1: The continuous assessment will mainly be developed in the context of small groups or computer lab sessions, with problem solving and exercises
N2: Partial written exam
N3: Final written exam
NT: Final project

$$N_{Final} = 0.20 \times N1 + 0.20 \times N2 + 0.40 \times N3 + 0.20 \times NT$$
Bibliography

**Basic:**


**Complementary:**


**Others resources:**

Hyperlink

Fortran

Maplesoft

North Western