Course guide
390334 - PRPE - Programming and Problem Solving in Bioengineering

Last modified: 06/06/2023

Unit in charge: Barcelona School of Agri-Food and Biosystems Engineering
Teaching unit: 749 - MAT - Department of Mathematics.

Degree: BACHELOR'S DEGREE IN BIOSYSTEMS ENGINEERING (Syllabus 2009). (Compulsory subject).

Academic year: 2023 **ECTS Credits:** 6.0 **Languages:** Catalan

LECTURER

Coordinating lecturer: Ginovart Gisbert, Marta

Others: Ginovart Gisbert, Marta

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.

Generical:
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), a mathematical software (Maple or some other), and a program for data analysis (R 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

Type	Hours	Percentage
Hours large group	40,0	26.67
Hours small group	20,0	13.33
Self study	90,0	60.00

Total learning time: 150 h

CONTENTS

INTRODUCTION TO PROGRAMMING AND PROBLEM SOLVING

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

Full-or-part-time: 50h
Theory classes: 12h
Laboratory classes: 8h
Self study : 30h

ALGORITHMS AND DEVELOPMENT OF PROGRAMS

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

Full-or-part-time: 50h
Theory classes: 15h
Laboratory classes: 5h
Self study : 30h

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.
Introducction to R software for data analysis (or some other) with the use of some of the specific packages to perform manipulation, descriptive and inferential analysis, and fitting and modeling of data sets, with special emphasis in the programming code it uses (through the RStudio interface).
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

Full-or-part-time: 50h
Theory classes: 13h
Laboratory classes: 7h
Self study : 30h

ACTIVITIES

(ENG) ACTIVITAT 1: CLASSES D'EXPLICACIÓ TEÒRICA

ACTIVITY 2: INDIVIDUAL WRITTEN EXAMS

Description:
Individual assessment by individual written exam in classroom or computer lab. There will be one mid-term test 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.

Material:
Exam sheets and calculator, and where appropriate, specific software and some documentation.

Delivery:
Resolution of the test by the student. Once corrected, the students can check their corrected exams with the teacher during the hours stipulated for the revision.

Full-or-part-time: 2h
Theory classes: 2h

(ENG) ACTIVITAT 3: RESOLUCIÓ D'EXERCICIS I PROBLEMES

Description:
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.

Specific objectives:
At the end of such activities 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 sotfware to solve diferent types of problems.

Material:
Documentation of the activity available in Atenea and/or a printed copy, and specific software.

Delivery:
Students may deliver a report of the activity, can be evaluated immediately at the end of the activity through a questionnaire, or not directly, through written tests on the subject. In Atenea they will find the answers.

Full-or-part-time: 35h
Laboratory classes: 20h
Self study: 15h

ACTIVITY 4: FINAL PROJECT

Description:
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.

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.

Material:
Documentation of the activity available in Atenea and specific software.

Delivery:
Evaluation of the documentation generated.

Full-or-part-time: 15h
Self study: 15h

GRADING 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

NFinal = 0.20 N1 + 0.20 N2 + 0.45 N3 + 0.15 NT

BIBLIOGRAPHY

Basic:
- Goldschlager L, Lister A. Computer science: a modern introduction. 2nd ed. New Jersey: Prentice-Hall International, 1988. ISBN 0131659456.
- Railsback SF, Grimm V. Agent-Based and Individual-Based Modeling: A practical introduction. Princeton University Press, 2011. ISBN 9780691136745.
- Joyanes Aguilar, Luis; Rodríguez Baena, Luis; Fernández Azuela, Matilde. Fundamentos de programación : libro de problemas. 2a ed. Madrid [etc.]: McGraw Hill, cop. 2003. ISBN 8448139860.
- Joyanes Aguilar, Luis. Fundamentos de programación : algoritmos, estructuras de datos y objetos [on line]. 4ª ed. Madrid [etc.]: McGraw-Hill, cop. 2008 [Consultation: 23/07/2022]. Available on: https://www-ingebook-com.recursos.biblioteca.upc.edu/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=10211. ISBN 9788448161118.
- Wilensky, Uri. An Introduction to agent-based modeling : modeling natural, social, and engineered complex systems with netlogo. Cambridge (Mass.): MIT Press, 2015. ISBN 9780262731898.

Complementary:
- Shiflet AB, Shiflet GW. Introduction to computational science : modeling and simulation for the sciences. 2nd ed. Princeton, N.J. [etc.]: Princeton University Press, 2014. ISBN 9780691160719.
- Brassard G, Bratley T. Fundamentos de algoritmia. Madrid: Prentice Hall, 2008. ISBN 848966000X.
- Lucas M, Peyrin JP, Scholl PC. Algorítmica y representación de datos. Barcelona: Masson, 1985. ISBN 8431103639.
- Scholl PC, Peyrin JP. Esquemas algorítmicos fundamentales: secuencias e iteración. Barcelona: Masson, 1991. ISBN 84310550X.
- Aho AV. Estructuras de datos y algoritmos. México: Addison-Wesley Iberoamericana, 1988. ISBN 0201640244.
- Ellis TMR. Fortran 90 programming. Wokingham: Addison-Wesley, 1994. ISBN 0201544466.
- Peña Marí R. Diseño de programas: formalismo y abstracción. 3a ed. Madrid: Prentice Hall, 2005. ISBN 8420541915.
- Smith PD. Files and databases: an introduction. Addison-Wesley, 1987. ISBN 0201107465.

- Tremblay JP, Bunt RB. Introducción a la ciencia de las computadoras: enfoque algorítmico. McGraw-Hill, 1990. ISBN 9684513607.
- Chapman SJ. Fortran 95/2003 for scientists and engineers. 3rd ed. Boston: McGraw-Hill, 2008. ISBN 9780071285780.

RESOURCES

Hyperlink:
- Fortran
- Maplesoft
- North Western