Universitat Autònoma de Barcelona

2023/2024

Informatics and Programming Tools

Code: 106807
ECTS Credits: 6

Degree	Type	Year	Semester
2504602 Nanoscience and Nanotechnology	FB	1	1

Contact

Name: Xavier Cartoixa Soler

Email: xavier.cartoixa@uab.cat

Teaching groups languages

You can check it through this link. To consult the language you will need to enter the CODE of the subject. Please note that this information is provisional until 30 November 2023.

Teachers

Catalina Coll Benejam

Pedro Alcázar Guerrero

Prerequisites

None.

Objectives and Contextualisation

• Getting acquainted with the use of various computer tools for data processing and the graphic presentation of information.

• Knowing the basic structures of a program: types, branches, loops; as well as the phases of its creation.

• Being able to use the python language to perform common tasks in a Nanoscience and Nanotechnology laboratory.

Learning Outcomes

- CM12 (Competence) Solve problems in the field of nanoscience by selecting the most appropriate computer and programming tools.
- KM18 (Knowledge) Recognise the computer tools used for data processing, analysis and representation.

- KM19 (Knowledge) Identify the different stages of an executable program, from analysis to verification, and the tools available for each one.
- SM18 (Skill) Write simple analysis and problem-solving computer programs in different scientific programming languages.
- SM19 (Skill) Use computer applications to visualise, process and represent data.
- SM19 (Skill) Use computer applications to visualise, process and represent data.
- SM20 (Skill) Convey relevant scientific data by producing high-quality figures.
- SM20 (Skill) Convey relevant scientific data by producing high-quality figures.

Content

1. Configuration of the computing environment

1. WSL2, VMs, Cygwin, Dual boot

2. Software installation

3. python configuration

2. Familiarization in Linux environments (PAUL)

1. The terminal window

2. System configuration

3. Algorithms and basic structures

1. Basic blocks

4. python

1.Hello world

2. If, then, else

3. While, do while

4. For-loop

5. Functions and subroutines

6. Modules

7. Type of variables

8. Objects

5. Graphic presentation of information

1.Excel

2.gnuplot

3.matplotlib

6. Data processing

1.NumPy and SciPy

2. Numerical integration

3. Linear algebra

4. Fourier series

5. Interpolation of points

7. Classifications of programming languages

1. Functional vs Object Oriented (OO)

2. Compiled vs interpreted

3. Pass by value vs pass by reference

4. Type of a variable

8. Tools

1. Compilers and interpreters

2. Languages: Hello world in

3. Debuggers: gdb, idb, GUIs

4. IDEs: Eclipse, Visual Studio, kdevelop

5. Profilers and memory leak detectors

6. Online resources: repositories, documentation, stackoverflow

9. Final considerations

Methodology

Teaching will be based on theory lectures with sporadic use of the computer, complemented by problem sets with intensive use of the computer and laboratory practices where the contents learned will be applied to the control of instrumentation and analysis and visualization of data .

Autonomous activities will be carried out that will include the development of simple computer programs.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Laboratory practice	7	0.28	KM18, KM19, SM18, SM19, SM20, KM18
Lecture	30	1.2	CM12, KM18, KM19, SM18, SM19, SM20, CM12
Problem sets	15	0.6	CM12, SM18, SM19, SM20, CM12

Type: Autonomous			
Preparation of laboratory practice	15	0.6	
Study and programming	77	3.08	CM12, KM18, KM19, SM18, SM19, SM20, CM12

Assessment

The completion of laboratory practices is mandatory, and it is necessary to pass the labs separately.

To pass the subject, a minimum grade of 4 is required in the synthesis test. This can be obtained either:

a) When the average of the partial synthesis tests reaches 4, and none of the partial tests has a grade lower than 2.

b) When the recovery synthesis test reaches the minimum of 4.

To take the recovery synthesis test, it is necessary to have previously taken at least one of the partial synthesis tests, and have passed the labs.

"Matricula d'honor" will be awarded preferentially according to the results of the partial synthesis tests, over those of the recovery test. It will be possible to go to the recovery synthesis test to improve the grade, but in case of obtaining a grade lower than the average of the partial tests, the final synthesis grade will be the mean between the average of the partials and the final grade of the recovery.

This subject does not contemplate the single evaluation system.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Laboratory practice	25%	0	0	CM12, KM18, SM18, SM19, SM20
Problem sets and independent study	20%	0	0	KM18, KM19, SM18, SM19
Synthesis test	55%	6	0.24	CM12, KM18, SM18

Bibliography

- Eric Matthes, Python Crash Course : a hands-on, project-based introduction to programming, No Starch Press, San Francisco, 3rd Ed, 2023.
- Sébastien Chazallet, Python 3 : Los fundamentos del lenguaje, ENI Ediciones, 2ª ed, 2016.
- Connor P. Milliken, Python Projects for Beginners, Apress, 1st ed, 2020.
- Joel Grus, Data Science from Scratch : First Principles with Python, O'Reilly, Sebastopol, CA, USA, 2nd ed, 2019.

Software

The course will make intensive use of the python programming language, as well as sporadic use of other programs and languages. Assistance will be offered to set up the environment.