COURSE DATA

DATA SUBJECT

Code: 34241
Name: Fundamentos de programación para la física
Cycle: Undergraduate Studies
ECTS Credits: 6
Academic year: 2026-27

STUDY (S)

Degree	Center	Acad. year	Period
1105 - Degree in Physics	Facultat de Física	1	First quarter

SUBJECT-MATTER

Degree	Subject-matter	Character
1105 - Degree in Physics	Information technology	BASIC

COORDINATION

FUERTES SEDER ARIADNA

GRIMALDO MORENO FRANCISCO

SUMMARY

This is a subject of 6 ECTS, taught during the first semester of the first year of the degree program. Therefore, it does not require prior knowledge from other courses in the program, but it is related to the rest of the subjects, as it is intended to lay the foundations for students to understand the capabilities of computers as tools for computation, simulation, and analysis of experimental data in Physics. This will enable them to use computers for problem-solving, being aware of their potential applications as well as their limitations.

The objective of this course is to provide students with a basic background in programming, oriented towards its application in Physics, so that computational thinking is used both as a methodology and as a set of tools that allow them to tackle increasingly complex problems later on. These problems may be approached analytically or numerically, particularly in the modeling of physical laws, simulation of physical systems, and analysis of experimental data obtained in the various laboratories of the Physics degree.

Therefore, the aim is for students to acquire a sufficient understanding of algorithm design using structured programming, as well as the fundamental data structures, so that they can learn to use computers as tools for solving physical problems and illustrating mathematical concepts.

Regarding the practical component, the course aims to develop students' skills in writing programs using a widely-used, general-purpose structured programming language. To achieve this, students will be given exercises involving the design and implementation of various algorithms and programs that solve problems in Physics and numerical computation. In doing so, they will acquire the basic knowledge required to use symbolic computation tools and numerical methods that will be necessary in other courses of the Physics degree.

PREVIOUS KNOWLEDGE

RELATIONSHIP TO OTHER SUBJECTS OF THE SAME DEGREE

There are no specified enrollment restrictions with other subjects of the curriculum.

OTHER REQUIREMENTS

It is advisable to have experience of using personal computers, operating systems and basic software such as a word processor or spreadsheet.

COMPETENCES / LEARNING OUTCOMES

1105 - Degree in Physics

Ability to collect and interpret relevant data in order to make judgements.

Learning ability: be able to enter new fields through independent study, in physics and science and technology in general.

Modelling & Problem solving skills: be able to identify the essentials of a process / situation and to set up a working model of the same; be able to perform the required approximations so as to reduce a problem to an approachable one. Critical thinking to construct physical models.

Prob. solving and computer skills: be able to perform calculations independently, even when a small PC or a large computer is needed, including the development of software programmes.

Problem solving: be able to evaluate clearly the orders of magnitude in situations which are physically different, but show analogies, thus allowing the use of known solutions in new problems .

Students must be able to apply their knowledge to their work or vocation in a professional manner and have acquired the competences required for the preparation and defence of arguments and for problem solving in their field of study.

Students must be able to communicate information, ideas, problems and solutions to both expert and lay audiences.

Students must have acquired knowledge and understanding in a specific field of study, on the basis of general secondary education and at a level that includes mainly knowledge drawn from advanced textbooks, but also some cutting-edge knowledge in their field of study.

Students must have developed the learning skills needed to undertake further study with a high degree of autonomy.

Students must have the ability to gather and interpret relevant data (usually in their field of study) to make judgements that take relevant social, scientific or ethical issues into consideration.

To know how to apply the knowledge acquired to professional activity, to know how to solve problems and develop and defend arguments, relying on this knowledge.

DESCRIPTION OF CONTENTS

1. Introduction to Computers and Programming in Physics

Basic concepts.

Internal structure of the computer: control unit, arithmetic logic unit, storage unit, input unit, and output unit.

Programming languages and paradigms: procedural languages and declarative languages.

Operating system.

Use of programming in Physics

2. Programming in High-Level Languages

Algorithms.

Characteristics of high-level programming languages: Objects and references, Simple data types, strings and lists, and data input and output.

Phases in program creation: Problem analysis, Algorithm design, and Algorithm programming. Examples of problems in Physics.

3. Structured programming

Theorem of structured programming.

Design of structured programs. Control structures: Sequential structure, Conditional structure, and Iterative structure.

4. Files

Basic file concepts: Access types, Logical and physical files, and Binary and text files.

File processing.

5. Modular programming

Module definition: Modular programming, Subprogram definition: Functions, Subprogram parameters, and Identifier scope.

Recursion.

Introduction to computer-based solving of moderately complex physics problems.

6. Structured data types

String and list processing.

Collections.

Introduction to classes.

7. Computer science lab exercises

Introduction to the work environment.

Program construction. Types, Arithmetic Expressions, and I/O

Conditional Structures

Iterative Structures

Files

Functions

Strings

Lists

Records

Lists of Records

WORKLOAD

PRESENCIAL ACTIVITIES

Activity	Hours
Theory	30,00
Laboratory	30,00
Total hours	**60,00**

NON PRESENCIAL ACTIVITIES

Activity	Hours
Attendance at other activities	0,00
Individual or group project	40,00
Independent study and work	0,00
Preparation of lessons	30,00
Preparation for assessment activities	20,00
Resolution of case studies	0,00
Total hours	**90,00**

TEACHING METHODOLOGY

Contact teaching 40%:

Theoretical and practical classes: It addresses the conceptual and formal matter and resolution of problems or cases as the application of theoretical concepts. They are based mainly on lectures and the use of dialogic teaching tools as a graphical representation of solutions, design presentations, spreadsheet programs, etc.

Laboratory sessions in the computer room: standard computer package management, implementation and execution of programs in which simple algorithms are coded. Resolving doubts in dealing with theoretical concepts to problem solving and program implementation. Reinforcing aspects that are more difficult, and verification of student progress in the field, associated with a component of continuous assessment.

Student's personal work 60%:

- Study of the theoretical concepts.

- Problem solving, multiple choice questions, and works (individually or in groups).

- Performing exercises using computer interpretation, conclusions and implementation of memory for communication.

- Individual tutorials: querying of the teacher on student concerns and difficulties encountered in the study and resolution of problems, or discussion on topics of interest, bibliography, etc.

EVALUATION

The assessment system is as follows:

Students are expected to do 2 types of work:

1. Autonomous work and self-learning

2. Supervised work

1.- Autonomous work and self-learning will consist of activities done outside the classroom. The lecturer will guide this type of activities (reading, problem resolution, research, etc.) but students will not be graded, although they could ask the lecturer for their revision in the office hours.

2.- Supervised work will consist of activities proposed by the lecturer and they will be assessed in order to evaluate student evolution. The type of work will be practical exercises to be solved at the laboratory.

The main characteristics of these activities are:

- They will be evaluated by the lecturer.
- They will have a deadline or will be made in-person.
- They are mandatory.

These works will include a set of individual objective tests consisting of solving a simple problem by programming an algorithm in a computer (N_p_presencial).

Also, the students will have a final mark related to the practical exercises carried out in the laboratory and to a proposed final project. (N_continuous).

At the end of the course, the student will have to pass an exam that will cover both the theoretical and practical knowledge. This exam will evaluate, on the one hand, the degree of understanding of the theoretical aspects and the associated formalisms, both in general terms and applied to simple use cases. On the other hand, it will evaluate the ability of the students to solve problems by applying the formalism, as well as their critical capacity towards the obtained results. A proper argumentation and an adequate justification will be assessed in both parts. (N_exam).

At the first call, the final mark will be calculated by using this formula:

Final Mark = 35% N_p_presential + 15% N_continuous + 50% N_exam

It will be necessary to get more than 4 in the exam to be able to average and more than 5 as final grade to pass.

Final grade will be according to the current legislation (Consejo de Gobierno de la UVEG 27th of January of 2004, Reales Decretos 1044/2003 and 1125/2003):

From 0 to 4,9: Insufficient

From 5 to 6,9: Sufficient

From 7 to 8,9: Good-Very Good

From 9 to 10: Excellent or Distinction

At the second call, the grade of a final exam will only be considered. This final exam will consist of a written examination test and also with a practical test at the laboratory (50% each one). When agreed by the lecturer, it might also be possible to take into consideration the grades obtained in the already passed exercises at the first call.

REFERENCES

Basic:

- [Kent D. Lee (2014)] Python Programming Fundamentals (Springer). https://link.springer.com/book/10.1007/978-1-4471-6642-9
- [A. Marzal, I. Gracia, P. García (2014)] Introducción a la programación con Python 3. (Col·lecció Sapientia 93 www.sapientia.uji.esISBN: 978-84-697-1178-1) https://repositori.uji.es/items/992c7ee2-fef4-4061-9958-aefe932fd611
- [A. Downey, J. Elkner, C. Meyers (2002)] Aprenda a Pensar Como un Programador con Python (Green Tea Press). Traducido por M.A. Vilella, A. Arnal, I. Juanes, L. Amurrio, E.

Andia, C. Ballardini. https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf

Additional:

- [Kent D. Lee, Steve Hubbard (2015)]; Data Structures and Algorithms with Python. Undergraduate Topics in Computer Science, Springer Verlag. https://link.springer.com/book/10.1007%2F978-3-319-13072-9
- [Benjamin Baka (2017)] Python Data Structures and Algorithms. Packt Publishing https://ebookcentral.proquest.com/lib/univalencia/detail.action?docID=4868549