COURSE DATA

Data Subject	
Code	34241
Name	Informatics
Cycle	Grade
ECTS Credits	6.0
Academic year	2022 - 2023

Study (s)

Degree	Center	Acad. year	Period
1105 - Degree in Physics	Faculty of Physics	1	First term

Subject-matter

Degree	Subject-matter	Character
1105 - Degree in Physics	5 - Information technology	Basic Training

Coordination

Name	Department
FUERTES SEDER, ARIADNA	240 - Computer Science
GRIMALDO MORENO, FRANCISCO	240 - Computer Science

SUMMARY

This subject was considered of basic training lasting 6 ECTS quarterly located in the first year, first semester of the degree to which there is no prior relationship with other subjects of the degree but with other materials since it must lay the groundwork for the student to be able to use the computer in solving problems knowing their potential uses and limitations.

The aim of this subject is to provide students with a basic training in computer science as a tool that allows you to tackle increasingly complex problems later, both analytically and numerically, and analysis of experimental data obtained in different laboratories degree.

Therefore, it is that students get sufficient knowledge of algorithm design using structured programming, as well as fundamental data structures.

In regard to the practical, this course will seek to have students acquire skills development programs in a structured programming language and general purpose and widely used to acquire the basic knowledge to use computer tools symbolic and numerical methods needed in other areas of the 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.

OUTCOMES

1105 - Degree in Physics

- 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.
- Ability to collect and interpret relevant data in order to make judgements.
- 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 .
- 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.
- Learning ability: be able to enter new fields through independent study, in physics and science and technology in general.
- 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 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 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.
- Students must be able to communicate information, ideas, problems and solutions to both expert and lay audiences.
- Students must have developed the learning skills needed to undertake further study with a high degree of autonomy.

LEARNING OUTCOMES

Learning outcomes

- Having a basic knowledge about the internal structure of a computer at both the physical (e.g. CPU, memory...) and the logical level (e.g. operating system, programs...) so that you can understand how computers work internally.

- Having a general knowledge about the different types of programming languages (i.e. procedural, object oriented...).

- Using computer programming as a basic tool for scientific work, also knowing how numerical data can be digitally represented.

- Understanding the data types, variables, constants, control and data structures available in procedural programming languages.

- Using procedural programming to solve problems using a computer.

- Learning how to code simple algorithms by means of a structured programming language.

- Basic programming using a high level language (e.g. C, C + +...) and coding of the theoretical concepts (e.g. data types, variables, control structures, functions...) following a modular approach.

- Using operating systems and knowing their main features.

- Having a basic understanding of mathematical software suites of interest in physics (e.g. Matlab) and how to use them for the treatment of experimental data.

DESCRIPTION OF CONTENTS

1. Introduction

Concept of computer:Basic concepts.
Internal structure of the computer: Control unit, arithmetic logic unit, storage unit, input unit and output unit.
Languages and programming paradigms: procedural language and declarative language.
Operating system.

2. Algorithms and programs

Concept of algorithm.
Solving problems by algorithms.
Stages in conducting a program. Analysis of the problem.
Algorithm design: Modular programming and stepwise refinement.
Algorithm representation: pseudocode and organigramme.
Simple data types.
Control Structures: Sequential structure, Conditional structure, Iterative structure.

Modular programming.
Introduction to recursion.

3. Arithmetics and information representation in the computer

Numeration systems.
Conversion between different numeration systems: (binary, octal, hexadecimal y decimal)
Arithmetic and logical operations.
Information representation in the computer: alphanumeric data, integers and real numbers.
Floating point arithmetic.

4. Structured data types

Structured data concept.
Types of structured data.
Structure of continuous data: vectors, matrices, strings and records.
Pointers and dynamic data structures.
Introduction of linked linear structures: linked lists

5. Files

Files: definition and basics of.
Logical and physical files.
Processing files: making file, opened and closed. Reading and writing.

6. PRACTICAL SESSIONS

1.Introduction to the programming. Structure of a program.
2.Types of data: simple data, constants and variables. Arithmetic operators, input functions and output functions.
3.Relational operators and logical operators. Sequential control Structures
4.Iterative control Structures
5.Modular programming.Definition and inmplementation of fumtions. Library of functions.
6.Vectors and matrices
7. Strings and records.
8. Working with files.
9. To carry out a final program.
10. Introduction to use mathematical software packets helpful to Physics (Matlab)

WORKLOAD

ACTIVITY	Hours	% To be attended
Theory classes	30,00	100
Computer classroom practice	30,00	100
Development of group work	15,00	0
Development of individual work	25,00	0
Preparation of evaluation activities	20,00	0
Preparing lectures	15,00	0
Preparation of practical classes and problem	15,00	0
TOTAL	**150,00**	

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 work 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

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” o “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

- [A. Tucker, W. Bradley (1994)]. Fundamentos de informática. (MacGraw Hill).
- [W. Savitch (2000)]. Resolución de problemas con C++. El objetivo de la programación (Prentice-Hall)
- [L. Joyanes (2000)]. Programación en C++: Algoritmos, estructuras de datos y objetos (MacGraw Hill).

Additional

- [L. Joyanes, I. Zahonero (2001)]. Programación en C: Metodología, algoritmos y estructuras de datos (MacGraw Hill)
- [H.M. Deitel, P.J. Deitel (1995)]. Como programar en C/C++. (Prentice Hall).
- [F. Virgós Bel, J. Segura Casanovas (2008)]. Fundamentos de informática : [en el marco del espacio europeo de enseñanza superior] (McGraw-Hill/Interamericana de España)