COURSE DATA

Data Subject	
Code	34159
Name	Informatics
Cycle	Grade
ECTS Credits	6.0
Academic year	2023 - 2024

Study (s)

Degree	Center	Acad. year	Period
1107 - Degree in Mathematics	Faculty of Mathematics	1	First term

Subject-matter

Degree	Subject-matter	Character
1107 - Degree in Mathematics	7 - Information technology	Basic Training

Coordination

Name	Department
BENAVENT GARCIA, MARIA ROSER	240 - Computer Science
FERRIS CASTELL, RICARDO	240 - Computer Science

SUMMARY

In this subject, students learn knowledge basic of what it is a computer, which are their potential uses and their limitations, particularly referred the resolution of mathematical problems.

One will introduce to the student in the knowledge and handling of the operating system, as well as the description and the use of the network like fundamental part in the communication of information between computers and the remote work.

It is also tried to secure a sufficient knowledge of the design of algorithms by means of structured programming, as well as of the structures of fundamental data.

Concerning the practical part, in this subject we will deal with which the student as much strengthens the knowledge seen in the theoretical part in the knowledge of the computer as of the basic tools for the Internet use and he acquires abilities of development of programs in a programming language structured of general intention and extended use (C/C++)

The teachers of this subject are members of *Grup Consolidat d'Innovació Docent en Metodologies Docents Col.laboratives, Cooperatives i Competitives* and participate in the proposal "Xarxa d'Innovació Docent" with reference SFPIE_GER16_418250.

PREVIOUS KNOWLEDGE

Relationship to other subjects of the same degree

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

Other requirements

Ninguno.

OUTCOMES

1107 - Degree in Mathematics

- Capacity for analysis and synthesis.
- Capacity for organization and planning.
- Solve problems that require the use of mathematical tools.
- Ability to work in teams.
- Learn autonomously.
- Adapting to new situations.
- Apply the knowledge in the professional world.
- Expressing mathematically in a rigorous and clear manner.
- Reason logically and identify errors in the procedures.
- Capacity of abstraction and modeling.
- Participate in the implementation of software and learn mathematical software.
- Knowing the time and the historical context in which occurred the great contributions of women and men in the development of mathematics.
- Visualize and interpret the solutions obtained.

LEARNING OUTCOMES

- Internal structure of the computer, detailing the physical parts compose that it (CPU, memory, …). The logic parts that make it works too (operating system, programs, etc.).

-Definition of network of computers, tools and utilities for its use at the time of sharing information and working with remote computers.

-Concept of algorithm: Resolution of problems by means of algorithms. Analysis of the problem. Design of the algorithm (top-down design or modular and refinement by steps). Programming of the algorithm.

- Introduction of the basic structures of a high-level language: variables, constants, structures of control, modular programming, recursion, structures of data, files.

- Programming of algorithms in programming language C/C++

DESCRIPTION OF CONTENTS

1. Introduction to the computer science and the networks of computers

Basic concepts.
Internal structure of the computer: control unit, unit arithmetic-logic, unit of storage, unit of entrance and unit of exit.
Languages and paradigms of programming: procedural languages and declaratory languages.
Operating system.
Networks of computers. Utilities to share information.

2. Algorithms and programs

Concept of algorithm.
Resolution of problems by means of algorithms.
Analysis of the problem.
Design of the algorithm: top-down design or modular and refinement by steps.
Representation of algorithms: pseudocode and organizational charts or flow charts.
Simple data types.
Structures of control: sequential structures, repetitive structures, selective structures.
Modular programming.
Recursion.

3. Files

Files: definition and concepts.
Physical organization and logical organization.
Operations on files: creation, opening and close. Reading and writing.

4. Types and structures of data

Concept of structured data.
Structured data types.
Contiguous structures of data: vectors, matrices, chains of characters and structures (or registries).

WORKLOAD

ACTIVITY	Hours	% To be attended
Computer classroom practice	30,00	100
Theory classes	22,50	100
Other activities	7,50	100
Development of group work	10,00	0
Preparation of evaluation activities	22,00	0
Preparing lectures	5,50	0
Preparation of practical classes and problem	47,00	0
TOTAL	**144,50**	

TEACHING METHODOLOGY

The development of the subject structure in two sessions of theory to the week of one hour. In those sessions the theoretical concepts are introduced and next exercises appear to the student type that will be solved in class and that the student will take from reference to develop to the practical exercises and the seminaries.

Throughout the course, continuous assessment activities will be carried out: a control scheduled in the middle of the semester that will include the knowledge seen in class up to that moment, tests, exercises, workshops, interactive videos and, in addition, small surprise checks may be carried out at throughout the semester.

The student will have listings of additional exercises that will solve by their account to reinforce the acquisition of the contents seen in the theoretical classes.

Throughout the course seminary sessions will be planned. For these seminaries the accomplishment of works or reduced training exercises will set out that the students will raise in house and will realise and expose in class.

There will be a test in the middle of the semester, including the knowledge seen in class so far. In addition, they can be made small quizzes throughout the semester.

The laboratory sessions will be of 2 hours each session throughout the fourth month period. For these sessions, the students will have reviewed the main topics that are going to be used in the development of the practice and which they are in the statement of the bulletin of summarized practices. Also, the student

must have read and have included/understood the statements of the proposed exercises and to have reflected on the possible solution of the same. During the time of the practice, the students will solve the proposed exercises and will ask the professor those aspects of the exercises that do not understand.

The last session of practices the students will raise a programming project that will include the concepts learned during the course and in the sessions of previous practices and that will finish in the nonactual hours. This “final project” has to be sufficiently great and complex like so that it is necessary his modular decomposition. In addition, in this project all the concepts have to put themselves in practice that have been learning of individual way throughout the course.

EVALUATION

Throughout the course small exams will be realised that will include/understand both the knowledge seen in the theoretical classes and in the practical exercises. One scheduled in the middle of the semester and possibly some unscheduled. There will also be online tests, exercises, workshops and interactive videos. These activities are called continuous assessment activities.

Moreover, seminaries with accomplishment of works and oral presentations in group will be realised throughout the course that will be evaluated.

The students will have in addition a note of practices corresponding to the evaluation to the practices realised in the laboratory (10%) and of the work or proposed final project (90%). When finalizing the course will be realised a written examination that will include both the theoretical and practical knowledge. In first call, the final mark of the subject will turn out to weigh: 15% of the marks of the continuous assessment activities (exams, tests, interactive videos) realised during the course with 50% of the obtained note in the final examination, 10% of the marks obtained in the seminaries; and 25% rest of the marks of practices.

In second call, the notes of the parts approved in first call with the same consideration will be considered (continuous assessment activities during the course 15%, final examination 50%, and seminaries 10% and practices 25% whenever it is considered that the student has worked actively within the group to obtain that note). Students who fail in the first call can improve the mark of practices again giving the final project improved or to improve the mark of the final examination in the second exam call. Neither the marks of the seminaries, nor the marks of the continuous assessments during the course are possible to improve it for the second call.

Is mandatory to attend to the seminar sessions. Students cannot pass if they have not been attended at least an 80% of the total of the practical sessions. They must give in the final project, as well as, to obtain, at least a 4.5 in the final exam and in the practices.

Copies:

Any copy any part of any of the activities of the course will be a zero in the full activity (seminar, practice, control, ...). Detection of two copies in different activities will suspend the subject in both the first and second call. Will apply the same criteria to both the original and the copy.

All the above measures will be applied irrespective of the disciplinary procedure that the student may initiate and, if applicable, the sanction that proceeds in accordance with current legislation

REFERENCES

Basic

-

-Referencia b1: Apuntes de la asignatura

-Referencia b2: [W. Savitch (2000)]. Resolución de problemas con C++. El objetivo de la programación (Prentice-Hall)

-Referencia b3: [L. Joyanes (2000)]. Programación en C++: Algoritmos, estructuras de datos y objetos (McGraw Hill)

-Referencia b4: [F. Virgos (2008)] Fundamentos de Informática (en el marco del espacio europeo de enseñanza superior) McGraw Hill 2008. Ferran Virgos/ Joan Segura

Additional

-

-Referencia c1: [George Beekman (2005)]. Introducción a la informática (Prentice Hall)

-Referencia c2: [L. Joyanes, I. Zahonero (2001)]. Programación en C: Metodología, algoritmos y estructuras de datos (McGraw Hill)

-Referencia c3: [H.M. Deitel, P.J. Deitel (1995)]. Como programar en C/C++. (Prentice Hall)