

Degree	Type	Year
2503758 Data Engineering	FB	1

## Contact

Name: Eduardo Cesar Galobardes

Email: eduardo.cesar@uab.cat

## Teaching groups languages

You can view this information at the [end](#) of this document.

## Prerequisites

This is a first-semester-first-year course, therefore it has no prerequisites.

A laptop is recommended.

## Objectives and Contextualisation

This subject is general and introductory to computer science. It is divided into two main axes, the first comprises the study of the Operating System at user level and, especially, the ability to develop shell scripts in Linux and the basic learning of a compiled language (C). The second consists in the introduction of the basic concepts of the structure of the computers and their design principles.

Therefore, the general objectives that are proposed for the subject are the following:

- Provide an overview of information technology, introducing its historical background and explaining the basic concepts: hardware, software, operating system, computer structure, algorithm, program and programming languages, compiler, interpreter, etc.
- Familiarize the student with the computer and system software.
- Introduce the student in a compiled programming language (C).
- Introduce the student in the Linux OS and develop command programs in this operating system.
- Provide a vision of the functional units of the computer and its interconnection.
- Offer the student a broad knowledge of the low level computer performance.
- Understand how information is represented on the computer at the hardware level.
- Introduce machine code and assembler language and show the translation of the basic structures of high level languages to machine level.
- Show the different levels of hardware and software necessary for the operation of computers and their interconnection.
- Introduce the concepts of computer cluster and cloud computing.

## Competences

- Demonstrate sensitivity towards ethical, social and environmental topics.

- Design efficient algorithmic solutions to computational problems, implement them in the form of robust software developments which are structured and easy to maintain, and verify their validity.
- Make a critical evaluation of work carried out.
- Plan and manage the available time and resources.
- Search, select and manage information and knowledge responsibly.
- Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.

## Learning Outcomes

1. Analyse, design, distribute and maintain applications, ensuring their quality and maintainability.
2. Demonstrate sensitivity towards ethical, social and environmental topics.
3. Explain the scope of informatics.
4. Make a critical evaluation of work carried out.
5. Plan and manage the available time and resources.
6. Search, select and manage information and knowledge responsibly.
7. Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.
8. Use operating systems, databases and programmes commonly used in engineering, at user level.

## Content

- Introduction to system software. Code editors, programming environments, repositories.
- Operating system at user level. Introduce the Linux OS, its most important commands, the shell script. Making particular emphasis on its capabilities to handle data files.
- Introduction to compiled language C. Differences between interpreted and compiled languages, the basic structure of a C program and the most relevant data and programming structures.
- Representation of the information. Numeric information binary representation (integers, real), characters, logical information and other types of information (images and sound).
- Introduction to the structure of the computer. Functional units of the computer. Internal functioning of a computer. The processor execution flow. The input/output system. Concept of interruption. The memory system. Hierarchical structure of memory.
- Machine language and assembler. Address modes. Elementary didactic computer.
- Introduction to the structure and functions of the Operating System. Evolution. Management: processor, memory, I/O, files. System calls.
- Concepts about computer networks. Transmission of data. Computer networks. Clusters and Cloud Computing.

## Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Lab sessions (Development of lab exercises)	18	0.72	1, 2, 4, 5, 6, 7, 8
Problem solving seminars	11	0.44	4, 6, 7, 8
Theoretical lessons	36	1.44	3, 5, 6, 7, 8

Type: Supervised

Follow-up and reinforcement in the resolution of practical cases	10	0.4	4, 5, 6, 7, 8
Solving additional problems	15	0.6	5, 6, 7, 8
Type: Autonomous			
Autonomous study	62	2.48	2, 3, 5, 6, 7
Solving practical exercises and problems in group	51	2.04	1, 2, 4, 5, 6, 7, 8

In the development of this subject, three types of teaching activities can be differentiated:

- Theoretical classes. Explanation on the blackboard of the theoretical part of each topic of the program. The typical structure of a theoretical class of this type will be the following: in the first place an introduction will be made where the objectives of the explanation and the contents to be discussed will be briefly presented. In order to provide the appropriate context, the presentation will refer to the material exhibited in previous classes, so that the position of these contents is clarified within the syllabus of the subject. Next, the contents studied will be uncovered, including narrative expositions, formal developments that provide theoretical foundations, and intercalating examples, which illustrate the application of the exposed contents. The most relevant elements will be highlighted in such a way that it will be possible to distinguish its relevance from the peripheral aspects. Finally, the concepts introduced will be summarized and the conclusions will be elaborated, including an assessment of the extent to which the objectives proposed at the beginning of the lesson have been achieved.
- Practical classes. All the topics will be accompanied by a relation of problems that the student must try to solve. In this sense, and as the student progresses and deepens his/her knowledge, these problems will be gradually more complex, allowing in this way to clearly appreciate the advantages of using the methodological tools taught during the course. Those that the professor considers of greater interest or in which the students find greater difficulty will be corrected on the board. Before each session of problems the professor can propose a list of exercises that students will have to solve before delivering them.
- Classes in the laboratory. The practical part of each topic will be completed with at least one session in the laboratory, where the student will have to try to solve a specific problem. Some of these exercises must be delivered after class.

The management of the subject will be done through its moodle website on the Virtual Campus, which will be used to view the materials, manage the lab groups, make the corresponding deliveries, see the marks, communicate with the professors, etc.

Transversal competences.

The transversal competences that will be worked on and evaluated throughout the course are T02 - Find, select and manage information and knowledge responsibly, T06 - Critically evaluate the work done, T07 - Demonstrate sensitivity to ethical issues, social and environmental issues and T08 - Plan and manage the time and resources available. Work will be carried out during the sessions of the different types of planned activities and will be evaluated mainly in the evaluation tests (T02), the delivery and correction of problems (T02, T06 and T08) and in the delivery of the lab exercises ( T02, T06, T07 and T08).

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.

## Assessment

## Continous Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Lab sessions: 1st delivery	25% of the lab mark	2	0.08	1, 2, 4, 5, 6, 7, 8
Lab sessions: 2nd delivery	25% of the lab mark	2	0.08	1, 2, 4, 5, 6, 7, 8
Lab sessions: 3th delivery	25% of the lab mark	2	0.08	1, 2, 4, 5, 6, 7, 8
Lab sessions: 4th delivery	25% of the lab mark	2	0.08	1, 2, 4, 5, 6, 7, 8
Labs: validation test	The test has no weight in the evaluation, but must be passed to obtain the grade of the labs	2	0.08	1, 7, 8
Problems	100% of the problems part	6	0.24	3, 4, 5, 6, 7, 8
Theory: 1st Test	50% of the thoretical part	2	0.08	3, 6, 7
Theory: 2nd Test	50% of the thoretical part	2	0.08	3, 6, 7
Theory: Reassessment	100% or 50% depends on whether both tests are recovered or only one. Only for students who have not passed one or both tests	2	0.08	3, 6, 7

There are three types of activities in the subject:

- Theoretical lessons (T)
- Proposed exercises resolution (P)
- Development of laboratory exercices (PL)

These three activities are evaluated independently and the weighted sum of them will give the final grade of the subject: 45% theory + 20% problems + 35% laboratory.

The part of Theoretical lessons (T) will be evaluated with individual written tests. There is a first exam corresponding to the contents of the topics: SO at the user level, introduction to programming in C, Representation of Information. The second exam corresponds to the contents of the Introduction to the Structure of the Computer, Introduction to the Structure of the OS and Interconnection. Each exam has a weight of 50% of the theory mark. If you do not this part, there will be a recovery on the day that we have assigned in the recovery week in February, where the tests that have not been passed can be recovered. The minimum grade to pass each test is 5.

The part of Exercise resolution (P) is carried out in continuous group evaluation. The final grade is the weighted sum of the fixed deliveries that are requested in class. By its nature, this part is not recoverable.

The part of laboratory exercices (PL) is evaluated in a group manner. There will be 4 deliveries corresponding to the part of the OS at user level (Shell Scripting), programming in C, introduction to the structure of the computer (assembler) and system calls. Each part has a weight of 25% of the final mark, but also an individual validation test of all the practices will be carried out. To pass the laboratory exercices, the weighted mark of the 4 deliveries must be at least 5 and pass the validation test. Validation tests can only be recovered if the practical part is missed because of them, in case of suspending the practices it cannot be recovered.

If the theory or practices have a score of less than 5 but the weighted average of the block is equal to or greater than 5, the block grade will be 4.

For each assessment activity, a place, date and time of revision will be indicated in which the student will be able to review the activity with the professor. In this context, claims can be made about the activity mark, which will be evaluated by the professors responsible for the subject. If the student does not submit to this review, this activity will not be reviewed later.

Honor enrollments. Granting an honorific enrollment qualification is a decision of the faculty responsible for the subject. The regulations of the UAB indicate that MH can only be awarded to students who have obtained a final grade of 9.00 or more. It can be granted up to 5% of MH of the total number of students enrolled.

Recoveries: As described above, a student can recover the theory part. However, to have the right to recovery, it is mandatory to have attended both partial tests.

Repeaters: There will be no differentiated treatment for repeating students.

Use of AI tools (e.g. Chat GPT): The use of such tools will only be restricted in written tests (theory exams, problem quizzes, and practice validation tests). This means that it is imperative that you make critical use of these tools, that is, that you use them to learn, not to copy.

This subject does not offer the possibility of a unique/single evaluation.

Note about plagiarism:

Notwithstanding other disciplinary measures deemed appropriate, and in accordance with the current academic regulations, irregularities committed by a student that may lead to a variation of the qualification in an assessable activity will be qualified as zero (0). Assessment activities qualified in this way and by this procedure will not be recoverable. If it is necessary to pass any of these assessment activities to pass the subject, the student will directly Fail, without opportunity to recover it in the same course.

These irregularities include, among others:

- the total or partial copy of a practice, report, or any other evaluation activity;
- let others to copy;
- present a group work not done entirely by the members of the group (applied to all members, not only to those who have not worked);
- present as own materials prepared by a third party, even if they are translations or adaptations, and generally works with non-original and exclusive elements of the student;
- Have communication devices (such as mobile phones, smart watches, camera pens, etc.) accessible during theoretical-practical assessment tests (individual exams);
- Talk with classmates during the theoretical-practical assessment tests (exams);
- Copy or attempt to copy from other students during the theoretical-practical assessment tests (exams);
- Use or attempt to use written material related to the subject during the theoretical-practical evaluation tests (exams), when these have not been explicitly allowed.

The official numerical grade will be the lowest value between 3.0 and the average of the grades in case the student has committed irregularities in an evaluation activity (and therefore this subject could not be approved by compensation). In future editions of this subject, the student who has committed irregularities in an evaluation activity will not be recognized in any of the assessment activities carried out.

*Summarizing:* copy, let others copy or plagiarize (or attempt) in any of the assessment activities is equivalent to a FAIL, non-compensatory and without recognition of parts of the subject in subsequent courses.

## Bibliography

1. Fundamentos de los Computadores. Pedro de Miguel Anasagasti. Paraninfo. 9a edición. 2004.
2. Conceptos de Informática. Alberto Prieto Espinosa. Mc Graw Hill. 2005.
3. Fundamentos de informática y programación en C. Diego R. Llanos Ferraris. Paraninfo. 2010.
4. El Lenguaje de Programación C. Brian W. Kernighan, Dennis M. Ritchie. Prentice Hall. 2a edición. 1991.

## Software

In this subject we will work with the Linux operating system, so students can choose between installing a Linux distribution (preferably Ubuntu [<https://ubuntu.com/>]) on their computer or installing VirtualBox [<https://www.virtualbox.org/>] and create a virtual machine with Ubuntu or use the Windows subsystem for Linux installation (WSL2 [<https://docs.microsoft.com/en-us/windows/wsl/install-win10>]) to install Ubuntu directly on Windows 10 or 11.

Depending on the path chosen, some applications may need to be installed on the Linux machine (Git Client, C compiler, code editor, etc.).

## Language list

Name	Group	Language	Semester	Turn
(PAUL) Classroom practices	811	Undefined	first semester	morning-mixed
(PAUL) Classroom practices	812	Undefined	first semester	morning-mixed
(PLAB) Practical laboratories	811	Undefined	first semester	morning-mixed
(PLAB) Practical laboratories	812	Undefined	first semester	morning-mixed
(PLAB) Practical laboratories	813	Undefined	first semester	afternoon
(TE) Theory	81	Catalan	first semester	morning-mixed