

Foundations of Computer Science

Code: 107845
ECTS Credits: 6

2025/2026

Degree	Type	Year
Electronic Engineering for Telecommunications	FB	1
Telecommunication Systems Engineering	FB	1

Contact

Name: Joan Oliver Malagelada

Email: joan.oliver@uab.cat

Teachers

Marc Vallribera Ros

Marc Codina Barbera

Teaching groups languages

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

Prerequisites

Any one.

Objectives and Contextualisation

The objective of the Fundamentals of Computer Science subject is to introduce the basic concepts of computer science, the physical components of a computer, the basics of processor programming and the representation of information on the computer. Specifically,

- Learn the representation of information within a computer. This includes numbering systems (binary, decimal, hexadecimal), character encoding (ASCII, Unicode), and the representation of data such as integers, reals and booleans.
- Learn the basic components of hardware, such as the CPU, memory (RAM), storage devices (hard drives, SSD), and input and output peripherals. It explains how these elements interact and how they form part of a complete computer system.
- Learn the architecture of the processor and its programming.
- Introduce general concepts about operating systems and databases.

Learning Outcomes

1. KM04 (Knowledge) List the different sections that make up computers, as well as their internal functioning.
2. KM04 (Knowledge) List the different blocks that make up computers, as well as their internal functioning.
3. KM05 (Knowledge) Identify the different types of data and their storage in memory.
4. KM05 (Knowledge) Identify the different types of data and their storage in memory.
5. KM06 (Knowledge) Identify the different types of flow control of a computer program.
6. KM06 (Knowledge) Identify the different types of flow control of a computer program.
7. SM05 (Skill) Use the theoretical foundations of programming and programming languages for the development of software systems. Understand the basic architecture of the computer, its functional units and its theoretical foundations of programming.
8. SM05 (Skill) Use the theoretical foundations of programming and programming languages to develop software systems. Understand the basic architecture of a computer, its functional units and the theoretical foundations of programming.

Content

1. Introduction
2. Representation of data in the computer
 - Representation of information: text, numbers, images, etc.
 - Representation of characters. BCD, ASCII, Unicode
 - Decimal, binary and hexadecimal bases.
 - Representation of naturals and integers.
 - Representation of reals
3. Concepts of digital systems
 - Boolean functions and expressions.
 - Combinational components
 - Sequential components.
4. Processor and memory
 - CPU
 - Semiconductor memory.
 - RISCV-32I-simple architecture
5. Assembler. L'ISA RISC-V.
 - Simple RISCV-32I ISA.
 - Assembly programming.
6. Processor
 - RISCV-32I-simple architecture
7. Memory
 - Memory hierarchy.
 - Cache memory.
 - Concepts of virtual memory, pagination, segmentation.
8. Input/output
 - Architectures: I/O, DMA, interrupts.
 - Input interfaces,
 - RAID systems.
9. Introduction to the operating systems.

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			

Therorycal classes	36	1.44	KM04, KM05, KM04
Type: Supervised			
Class problems and laboratory sessions	24	0.96	KM06, SM05, KM06
Type: Autonomous			
Autonomous work	90	3.6	KM04, KM05, KM06, SM05, KM04

Theory classes:

Basic knowledge of the subject is given and indications on how to complete and deepen this content are given.

Problem classes:

The scientific and technical knowledge presented in the master classes is worked on. Problems are solved and practical cases are discussed. The problems promote the capacity for analysis and synthesis, critical reasoning, and the student is trained in the resolution of practical cases.

The methodology followed in problems is as follows: problem sheets are given, which the students must solve. In class, the doubts that have arisen are reviewed and those that the students have had conflicts with are resolved.

Eventually, in some problem sessions, group work is done to solve problems of subject synthesis with self-assessment.

Practices:

During the course, practical classes will be held. Students will work in groups of two.

In the practices, the student must develop the thinking habits specific to the subject and group work. The student will have to start solving algorithms.

The sessions may include self-assessment or evaluation exercises.

AI can be used as a learning tool. However, it is not permitted in the practical and assessment activities of the subject.

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
Continuous assessment tests	65% - 75%	0	0	KM04, KM05
Practical activities	25%	0	0	KM06, SM05
Supervised activities	10%-0%	0	0	KM04, KM05, KM06, SM05

This subject does not have a single evaluation system.

1. ABOUT THE TYPE OF EVALUATION ACTIVITY

Up to 3 types of activities are involved in the subject grade:

- Two CONTINUOUS EVALUATION tests. The final weight in the subject is between 65% and 75%, depending on the number of supervised evaluation activities that have been carried out.

- Evaluation activities proposed in the PRACTICAL CLASSES. The final weight in the subject will be 25%. Attendance is mandatory.
- SUPERVISED EVALUATION ACTIVITIES may be carried out during the course. The maximum weight of these activities will be 10%. The weight will depend on the total number of activities carried out.

2. CONTINUOUS EVALUATION TESTS

- There will be two continuous evaluation tests. The continuous evaluation dates are set at the beginning of the course by the coordination of the degree and do not have an alternative date for recovery in case of absence.
- The weight of each test will be 50%.
- The final grade of the continuous assessment tests is the arithmetic average of the two grades.
- To be able to take an average, you must have at least 3.5 (out of 10) in each of the two tests.

3. EVALUATION IN PRACTICAL ACTIVITIES

The practices are approved by continuous assessment.

The assessment activities may be:

- Submission of reports and/or exercises from the practice sessions. If there are, the grade will be of 1 point (out of 10) in total.
- Assessment tests for the practical class (they may be oral).
- You must have a 5 (out of 10) in the assessment of the practices to be able to take an average with the theory grade and pass the subject.
- You must take all the practical assessment tests and get more than 1.5 (out of 10) in each test to be able to take an average.

4. ABOUT SUPERVISED ASSESSMENT ACTIVITIES

- They may be done during the course as collections of continuous assessment activities in class.
- There is no requirement that affects the final grade.
- They cannot be recovered.
- If done, they may count for up to 10% of the final grade, which will be adjusted based on the grade of the continuous assessment activities.

5. ABOUT THE FINAL SCORE

Anyone is considered passed who

- has a grade equal to or greater than 5 (out of 10) (calculated according to the established scales) and
- has at least a 5 in the practical activities and
- does not have any grade in the continuous assessment below the minimum grade (which is 3.5 (out of 10)).

6. ABOUT POSSIBLE VALIDATION OF INTERNSHIPS:

At the beginning of the academic year, if possible, you will be notified if there is validation of internships. If applicable, practical classes must have been completed and passed the previous year..

7. Grades may be reviewed within the period established for each activity.

8. The MH grade will be awarded if 90% of the score is exceeded in the percentage established by the university and provided that the student demonstrates excellence in all their assessed activities. A non-assessable will be considered in the event that a student has not carried out any assessment activity. The failure for not having achieved the minimum grade in any of the assessment activities is the min{average grade, 4.5}.

9. Any change in the schedule will be reported to the Virtual Campus.

10. Without prejudice to other disciplinary measures deemed appropriate, and in accordance with current academic regulations, irregularities committed by a student that may lead to a variation in the grade will be graded with a zero (0). For example, plagiarizing, copying, allowing copying, ..., an assessment activity, will imply the suspension of this assessment activity with a zero (0). Assessment activities graded 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 subject will be directly suspended, with no opportunity to recover it in the same course.

Bibliography

Generic book for the course:

- Linda Null. "Essentials of Computer Organization and Architecture", Sixth Edition, Burlington, MA: Jones & Bartlett Learning, 2023. ISBN 978-1284259445

- D. Patterson, A. Waterman. "Guia práctica de RISC-V. El atlas de una arquitectura abierta." : ISBN 978-0-9992491-2-3. Document pdf. (Recomanat pel RISC-V).

Complementary bibliography:

Specific for the processor:

- S.L. Harris, D. Harris. "Digital Design and Computer Architecture, RISC-V Edition". Morgan-Kaufman. 2021

Specific for digital systems:

- J. Oliver, C. Ferrer. "Diseño de sistemas digitales: introducción práctica". Servei de publicacions de la Universitat Autònoma de Barcelona. 1998. ISBN 84-490-1445-X

Specific for operating systems:

- Pedro de Miguel Anasagasti, Fernando Pérez Costoya. "Sistemas Operativos", D. Arquitectura y Tecnología de Sistemas Informáticos, ETSE, UPM. 2016. Bajo licencia Creative Commons NoComercial CompartirIgual 4.0.

Software

Complex programming tools are not required for the development of the subject.

Network tools will be used for the introduction to assembly language programming.

Groups and Languages

Please note that this information is provisional until 30 November 2025. You can check it through this [link](#). To consult the language you will need to enter the CODE of the subject.

Name	Group	Language	Semester	Turn
(PAUL) Classroom practices	311	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	312	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	331	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	332	Catalan	first semester	morning-mixed
(PLAB) Practical laboratories	311	Catalan	first semester	morning-mixed
(PLAB) Practical laboratories	312	Catalan	first semester	morning-mixed
(PLAB) Practical laboratories	313	Catalan	first semester	morning-mixed
(PLAB) Practical laboratories	314	Catalan	first semester	morning-mixed

(PLAB) Practical laboratories	315	Catalan	first semester	morning-mixed
(PLAB) Practical laboratories	316	Catalan	first semester	morning-mixed
(PLAB) Practical laboratories	317	Catalan	first semester	morning-mixed
(PLAB) Practical laboratories	318	Catalan	first semester	morning-mixed
(TE) Theory	31	Catalan	first semester	morning-mixed
(TE) Theory	33	Catalan	first semester	morning-mixed