Universitat Autònoma de Barcelona

2020/2021

Foundations of Computer Science

Code: 103799
ECTS Credits: 9

Degree	Type	Year	Semester
2500895 Electronic Engineering for Telecommunication	FB	1	1
2500898 Telecommunication Systems Engineering	FB	1	1

The proposed teaching and assessment methodology that appear in the guide may be subject to changes as a result of the restrictions to face-to-face class attendance imposed by the health authorities.

Contact

Name: Joan Oliver Malagelada
Email: Joan.Oliver@uab.cat

Use of Languages

Principal working language: catalan (cat)
Some groups entirely in English: No
Some groups entirely in Catalan: Yes
Some groups entirely in Spanish: No

Teachers

Marta Prim Sabria
Joaquín Saiz Alcaine
Roger Malet Munté
Carlos Puig Toledo
Raimon Casanova
Vanessa Moreno Font
Marc Vallribera Ros

Prerequisites

There is not.

Objectives and Contextualisation

The subject corresponds to a basic subject of introduction to the foundations of computers and to the programming in the degrees of telecommunications. It is taught in the first semester.
The basic training objectives of the subject are:
- Introduce the student in the general concepts of the computer
- Learn the basic architecture of a generic processor and work with the basics of its low level operation.
- Learn programming methodology and apply it to the creation of programs in a high level language.
In this sense, the subject introduces the student into the concepts of the hardware of the computer. Also the subject introduces the students into the C programming.

Competences

Electronic Engineering for Telecommunication

- Develop personal work habits.
- Develop thinking habits.
- Learn new methods and technologies, building on basic technological knowledge, to be able to adapt to new situations.
- Work in a team.

Telecommunication Systems Engineering

- Develop personal work habits.
- Develop thinking habits.
- Learn new methods and technologies, building on basic technological knowledge, to be able to adapt to new situations.
- Work in a team.

Learning Outcomes

1. Critically evaluate the work done.
2. Define the architecture of computers and servers and use operating systems.
3. Define the architecture of computers and servers, and use of operating systems.
4. Describe, on a basic level, the use and programming of computers, operating systems, databases and computer programs that have applications in engineering.
5. Describe, on a basic level, the use and programming of computers, operating systems, databases and computer programs with engineering applications.
6. Design and implement databases in the development of software applications and, specifically, applied to information systems.
7. Develop independent learning strategies.
8. Develop scientific thinking.
9. Develop the capacity for analysis and synthesis.
10. Identify the functional units of a computer.
11. Use basic programming theory and language programming to develop software systems.
12. Use the basic theories of programming and programming languages to develop software systems.
13. Work autonomously.
14. Work cooperatively.

Content

A. FOUNDATIONS OF PROGRAMMING
1. Computer engineering and computing
2. Representation of information in the computer
3. Algorithms design
4. Introduction to C
5. Control structures
6. Functions and procedures
7. Vectors (arrays: lists and tables)
8. Structures (registers)
9. Files
B. COMPUTER BASICS
10. Digital components
11. Introduction to computer architecture
12. Machine language and assembler
13. Operating systems.Files and databases.

Methodology

Master classes:

Basic knowledge of the subject is given in class and indications of how to improve these contents.

Class problems:

The scientific and technical knowledge exposed in the master classes is worked on. Problems are solved and case studies are discussed. With the problems, the ability to analyze and synthesise, the critical reasoning is promoted, and the student is trained in solving problems.

The methodology followed in problems is the following: issues of problems are given, which students must solve. In class a review of the doubts that have arisen are commented and resolved.

Eventually, in some problem sessions, problems are collected and corrected.

Practices:

During the course there will be 6 practices of 3 hours. Students will work in groups of 2.

Te student will develop own thinking habits of the subject. The student will be introduced in the programming language.

Notes due to Covid-19.

1. The current circumstances make unpredictable to think about the normality with which teaching will take place during the first semester of the course.

Therefore, in almost total security, part of the teaching of laboratories will have to be taught in virtual mode.

2. On the other hand, the proposed teaching methodology and assessment may undergo some modification depending on the attendance restrictions imposed by the

health authorities.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Master classes	33	1.32	4, 8, 9, 6, 10, 12
Type: Supervised			
Class problems and laboratory classes	32	1.28	14
Type: Autonomous			
Autonomous work	150	6	1, 7, 13

Assessment

1. Evidence of continuous evaluation. There are two tests that include the three blocks of matter (A, B and C of the section contained)

Evidence of continuous evaluation	Weight note continuous assessment	Minimum mark to make an average

A. Programming foundations	50%	3.5
B. Computer basics	50%	3.5

2. Activities involved in the final evaluation note

Final note	Weight final mark (%)
Continued evaluation	65-75
Class marks	10-0
Practices	25

3. Anyone who is approved is considered approved
- have a final grade equal to or greater than 5 i
- Have approved practices (minimum 5) i
- there is no remaining block (A, B) of the subject of the continuous evaluation below the minimum mark (3.5) to do the average.
4. The practices are approved by continuous evaluation, and the note is a function of the preparation of the practice, the execution of the practice and the final report. The practices can not be recovered.
5. Class collections can not be retrieved.
6. There will be a final exam of the three blocks of theory aimed at recovering the unresolved part of the continuous evaluation. It is normative that attending this final test has assisted at least two of the three partial tests.
7. At the beginning of the academic year, if possible, it will be notified if there is validation of internships. In case of being, the validation of practices will only be carried out to those students that request it and have approved the practices in the previous year.
8. Continuous evaluation dates are set at the beginning of the course and do not have alternative recovery date in case of non-attendance. If there is any change in programming due to adaptation to possible incidents, the virtual campus will always be informed about these changes.
9. Notwithstanding other disciplinary measures deemed appropriate, and in accordance with the current academic regulations, irregularities committed by a student that can lead to a variation of the qualification will be classified by zero (0). For example, plagiarizing, copying, copying, ..., an evaluation activity, will imply suspending this evaluation activity with 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, this subject will be suspended directly, without opportunity to recover it in the same course.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Practical classes	25	4	0.16	1, 7, 14, 13
Supervised	10	2	0.08	8, 9, 14
Theory	65	4	0.16	2, 3, 4, 5, 6, 10, 12, 11

Bibliography

The bibliography that will be used in the subject will be:

Class books:

- J. Oliver. Introducció als Fonaments de computadors amb EduP12. Editat per CVC. 2012. http://www.cannic.uab.cat/llibre/EduP12_v1.pdf.
- A. Prieto, A. B. Prieto. Conceptos de informática. Edit Mc Graw Hill.2005.
- J. Antonakos, K. Mansfield. Programación estructurada en C. Edit Prentice Hall. 2000.
- G. J. Bronson. C++ para Ingeniería y Ciencias. Edit. International Thomson Editores. 2000.

Complementary Books:

- E. Quero. Programación en lenguaje C: ejercicios y problemas. Edit Paraninfo. 1998.
- J. Oliver, C. Ferrer. Disseny de sistemes digitals. Servei de Publicacions de la UAB. 1990.

Web pages:

- https://cv.uab.cat/. Website of the Virtual Campus of the UAB. It will serve as the base page for interaction with the subject and the notes and statements of problems and practices will be posted.
- http://www.cannic.uab.es/docencia/FI/FI.htm. Web page in which there is information about the subject of the subject.
- http://www.cannic.uab.es/docencia/Fp/FP.htm. Website where you can find notes and programming problems in C.