

Degree	Type	Year
2502441 Computer Engineering	FB	1

## Contact

Name: Francesc Xavier Roca Marva

Email: xavier.roca@uab.cat

## Teachers

Francesc Xavier Roca Marva

Miquel Àngel Senar Rosell

Sandra Adriana Mendez

Daniel Munzon Gil

Diego Alejandro Velazquez Dorta

Roberto Benavente Vidal

## Teaching groups languages

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

## Prerequisites

Since the subject is an introduction it will be assumed that there is no prior knowledge about the subject. It is the responsibility of the subject to provide students with a means to acquire the knowledge described in the contents section of the subject (section 6 of this guide).

However, it is advisable:

- having attended the technological baccalaureate,
- Have user-level knowledge of some type of platform (windows, mac or linux)
- have access to a computer, if it is portable better

## Objectives and Contextualisation

This subject is general and introductory in computer science. It is divided into two main axes: the first includes the study of the methodological aspects of programming and learning a high level language and the second consists in the introduction of the basic concepts of the structure of computers and principles of design. Therefore, the general objectives that are proposed for the subject are the following:

- Provide an overview of computer science, 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.
- Understand the life cycle of the software: analyze the problem (understand what is being asked), design (propose a solution to the problem), implementation (coding in a programming language of the chosen solution), test (completion of a test systematically to ensure the correctness of the implemented solution).
- Provide the student with the ability to design algorithms to solve problems, introducing a rigorous and structured programming methodology progressively and systematically, based essentially on the technique of downstream design of algorithms.
- Introduce the student in a real programming language. It is intended that the student perceives the difference between the flexibility of the pseudo-algorithmic notation that is used in the first subjects and the strict syntax of a real programming language, both in its lexical aspects (valid words of the language), syntactic (rules to combine) and semantic (meaning of the same).
- To habit the student to develop programs following a style norms that seek to achieve quality programs. These rules of style include those that facilitate the compression of the code, such as the use of comments, indentation of the code, the use of names suitable for data types, etc.
- Provide a vision of the functional units of the computer and its interconnection.
- Offer the student a broad knowledge of how the computer works at a low level.
- Introduce machine code and assembler language and show the translation of the basic structures of high level to low level.
- Show the different levels of hardware and software necessary for the operation of computers and their interconnection.

## Competences

- Acquire thinking habits.
- Have basic knowledge of the use and programming of computers, operating systems, databases and computer programs with applications to engineering.
- Have the capacity to understand and master the basic concepts of discrete mathematics, logic, computational algorithms and complexity, and their application to the resolution of engineering problems.
- Know about the structure, organisation, operation and interconnection of computer systems, basic programming, and the application of the same to solve engineering problems.
- Know the basic materials and technologies to enable the learning and development of new methods and technologies, as well as those that provide large-scale versatility to adapt to new situations.

## Learning Outcomes

1. Apply the basic principles of computer programming to solve engineering problems.
2. Develop a capacity for analysis, synthesis and prospection.
3. Know and apply the principles of system interconnection.
4. Know and be able to use operating systems, databases and programs that are commonly used in engineering.
5. Know the basic principles of the structure and programming of computers.
6. Recognise and identify the methods, systems and technologies of computer engineering.
7. Understand and apply basic knowledge of computer logic.

## Content

### BLOCK 1: INTRODUCTION TO COMPUTERS

## Unit 1: Introduction to computing

History, Functional structure of the computer, Programs / instructions, Conceptual levels of the computer.

## BLOCK 2: INTRODUCTION TO PROGRAMMING

### Unit 2: Problem solving: introduction to algorithms and programming.

Introduction to problem solving. Concept of algorithm. Phases in the development of algorithms. Programming as an engineering discipline. Software life cycle. Basic elements of an algorithm. Tools for the representation of algorithms. Programming languages. Classification Language Translators: Compilers and Interpreters.

### Unit 3: Basic concepts and control

structures Sequential structure. Selection structures. Iterative or repetitive structures.

### Unit 4: Subprograms. Procedures and functions

The concept of subprogram as an abstraction of operations. Location, level, scope and visibility. Definition of functions. You call functions. Declaration or prototype of functions. Types of arguments: input, output, input / output. Function step to functions: by value and by reference.

### Unit 5: Data types

Definition of variables and constants. Types of fundamental data. The Array type. One-Dimensional Arrays. Multidimensional Arrays. Chains of characters. Related functions. The type register. Records arrays.

## BLOCK 3: COMPUTING INTRODUCTION

### Unit 6: Representation of the information in the computer

Numbering systems: decimal binary, operability, octal and hexadecimal systems. Representation of numerical data (C1 / C2). Real numbers Representation of text.

### Unit 7: Introduction to the computer architecture

Functional units of the computer. Internal functioning of a computer. The processor Execution flow of instruction. The input / output system: Programmed Output / Output. Concept of interruption The memory system. Hierarchical structure of memory. Vision of a PC.

### Unit 8: Machine language and assembler

Direction. Elementary didactic computer Assembly language

### Unit 9: Peripherals.

### Unit 10: Operating system. Evolution. Management: processor, memory, I / O, files

### Unit 11: Concepts on computer networks

Data transmission. Computer networks. Internet

## **Activities and Methodology**

Title	Hours	ECTS	Learning Outcomes
-------	-------	------	-------------------

Type: Directed

Explanation and resolution of problems	7.5	0.3	1, 7, 4, 6
Laboratory practices	14	0.56	1, 5, 6
Presential class / Theory	42	1.68	7, 5, 3, 4, 6
Type: Supervised			
Additional problem-solving	15	0.6	1, 7, 6
Follow-up and reinforcement in the resolution of the practical cases proposed	10	0.4	1, 7, 5, 6
Type: Autonomous			
Individual study	62	2.48	7, 5, 3, 4, 6
Resolution of practical cases	28	1.12	1, 7, 6
Solving problems individually and in groups	22	0.88	1, 7, 5, 6

The subject has two different parts Programming and Computing. Each part will have a teaching methodology adapted to the educational contents.

### Programming

The teaching management of the subject will be carried out through the documentary manager Caronte (<http://caronte.uab.cat/>), which will be used to view the materials, manage the groups of practices, make the corresponding deliveries, to see the notes, to communicate with the teachers, etc. To be able to use it you have to do the following steps:

1. Register as a user giving the name, NIU, and a photo ID in JPG format. If you have already registered for another subject, you do not need to do it again, you can go to the next step.
2. Register for the "Teaching of Foundations of Computing" teaching type, giving as a subject code the one given the first day of the course.

In the development of the programming part, the reverse class methodology will continue. This means that the students will have to work some knowledge before each session to be able to leave for the face-to-face sessions those questions that can only happen face to face. The starting point is the experiences and knowledge of the participants themselves, who generate the conceptual framework in an interactive way.

Educational activity is configured as a double-track system where teachers and students build together and share responsibilities about the learning process where the objective is to see the maximum possible cases to understand the complexity of what it implies Raise a solution to a programming problem.

- Before Class. The basics work before face-to-face sessions. Prepare the class with the contents you will find one week in advance to the platform [Caronte](#). The available material will be written documentation, multimedia content and learning activities. In parallel to the material there will be a discussion forum in [Caronte](#), moderated by the teaching staff, to solve doubts and tutorial sessions. Each student can set their own pace by reviewing materials as many times as they need to acquire these concepts. By way of indication, each student should spend about 2 or 3 hours for preparation. There will be a questionnaire to evaluate the achievement of the concepts explained
- In class. The objective of face-to-face sessions is to move from information to knowledge. We will put the emphasis on creating knowledge applicable to the analysis of specific problems, in which the teachers test their students, which consolidate their learning from problems. The structure of the classes will be:  
Review of the basic concepts more important (15 min). The aim is to solve the doubts that may arise over theoretical aspects not understood and track the acquisition of the basic concepts.

Proposal of problems (rest of the class time) The objective is to experiment with a practical task that allows to apply the concepts learned to solve challenges and thus to share and create knowledge.

- After Class. The objective is to consolidate knowledge. In the platform [Caronte](#), a set of evaluable self-correction problems can be found that will allow students to deepen their understanding and personalize knowledge. The fact that it is self-assessed allows the pace of consolidation to be adjusted. The resolution should allow students to reflect on their own learning.

### Computation

The teaching management of the subject will be done through the documentary manager Campus Virtual (<http://cv.uab.cat/>), which will be used to view the materials, manage the groups of practices, make the corresponding deliveries, to see the notes, to communicate with the teachers, etc.

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

- Theoretical classes. Exhibition on the board of the theoretical part of each theme of the program. The typical structure of a master class of this type will be the following: in the first place an introduction will be made where the objectives of the exhibition 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 general framework 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 important elements will be highlighted in such a way that it is able to distinguish the relevance of 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 subjects will be accompanied by a relation of problems that the student must try to solve. In this sense, and as the student progresses in the depth of their 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 teacher can provide a list of exercises that students will have to solve before delivering.
- Classes in the laboratory. The practical part of each subject 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 resolution of these exercises should be used as a learning base to be able to carry out a practical case that will be proposed as work throughout the course.

### Transversal competences

The competence that will be worked on and evaluated throughout the course is T01.02 - Develop the capacity for analysis, synthesis and prospecting. It will shoot throughout the sessions of the different types of activities planned and will be evaluated mainly in the evaluation tests and in the delivery of the practices.

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
Block A: Continued assessment of the proposed problems	12% of the subject	4.5	0.18	1, 5, 2, 6

Block A: Delivery of the programming practice	18% of the subject	6	0.24	1, 5, 2, 6
Block A: Theoretical Individual Exams (partial)	30% of the subject	2	0.08	1, 5, 2
Block B: Continuous evaluation of the proposed problems	4% of the subject	3	0.12	1, 7, 4, 6
Block B: Delivery of the corresponding machine-level practice	12% of the subject	3	0.12	1, 5, 2, 6
Block B: Theoretical Exam Individual (partial)	24% of the subject	2	0.08	1, 5, 2, 6
Re-evaluation Theory Part A and / or B: Individual proof of Theory (only for students who have suspended part A and / or B)	30% corresponding to part A and 24% corresponding to part B	4	0.16	1, 7, 5, 3, 4, 6

**IMPORTANT:** This subject does not have "unique assessment" system.

#### **Planning and recovery of programmed evaluation activities**

The subject is divided into two blocks:

1. The first nine (9) weeks will be devoted to the study of The programming will count 60% of the final grade value (Block A).
2. The following six (6) weeks will be devoted to the study of the computer and will count 40% of the value of the final mark (Block B).

The two blogs will be evaluated independently. To obtain the final note the two parts must be approved separately (Note >= 5).

The final grade of the subject will be: 60% block A + 40% block B

#### **Programming (BLOCK A)**

The following assessable activities are carried out: grade

Activity	Date	Recovery	Percent Block	Minimum	Percentage Subject
Theoretical Exam (T) Individual	Check Planning	Consultation Planning	50 %	T >= 5	30%
Self-assessment questionnaire of achievement concepts	Weekly and until theory recovery exam	No	5%	No Serves to unlock access to problems	3%
Problems resolution (P) Individual	Up to theory recovery exam	No	15%	No	9%

It can only be accessed if the achievement questionnaire has been previously completed (some ones will have deadline)

Laboratory practice (PL) Group	Consult Planning  Compose of a test (PLt) and the correction of the final delivery (PLf)	Consult Planning	30%	PL = 30% PLt + 70% PLf  PLt and PLf >= 5	18%
--------------------------------	--	------------------	-----	--	-----

### Computation (BLOCK B)

The following activities are evaluated:

Activity	Data	Recovery	Percentage Block	Minimum Note	Percentage Subject
Theoretical Exam (T) Individual	Check Planning	Check Planning	60%	T >= 5	24%
Resolution Problems (P) Individual	Check Planning	No	10%	No	4%
Laboratory Practicum (PL) Grupa I	Check Planning  You have three deliveries, one for each full practice.  P1, P2 & P3	Check Planning	30%	PL = 33% P1 + 33% P2 + 33% P3  P1, P2, P3 >= 3.5 i PL >= 5	12%

Overall, for both blocks

Both the theory and practices (labs) parts must be passed with a grade of 5 or higher, regardless. Practices are mandatory as well as attendance at all sessions. To pass the labs you have to do it one at a time. Only in extraordinary cases will it be possible to access the re-evaluation of the labs, always at the discretion of the teaching staff of the subject.

The grade for each block is obtained from the weighted sum of the assessment activities with their corresponding percentage. The result of the weighted sum must be >= 5 to consider each block approved.

The Theory part (T) will be assessed with an individual written exam. There is a first individual exam corresponding to the first block (block A) and another to the second (block B). In the case of not passing any of these tests, there will be a recovery on the day we have assigned to the week of recovery in January/February where you can recover the tests that have not been passed in the partial exams. The minimum grade to pass each test is 5.

The dates of evaluation and delivery of assignments will be published in the document manager chosen by each block (Caronte (block A) or UAB Virtual Campus (block B)) and may be subject to schedule changes for

reasons of adaptation to possible incidents. S 'will report the changes through the document manager because it is understood that this is the usual platform for exchanging information between teachers and students.

#### Grade review procedure

Students will have the right to review the theoretical exams (T). The place, date and time of the review will be published on the day the notes are published. If the student does not appear for this review, it will not be reviewed later. Only unjustified cases can a review be made after the set date and always up to a maximum of 7 calendar days later.

The marks of the laboratory practice (PL) will not be reviewed because the assessment is done in front of the students.

#### Qualifications

Final grade: Both blocks will be evaluated independently. In order to obtain the file grade, the two blocks must be approved separately (Grade  $\geq 5$ ).

The grade for each block comes from the weighted sum according to the criteria set out in the assessment activities section. If the theory (T) or the practices (PL) have a grade lower than 5, the grade will be based on the following formula: Minimum (apply weighted sum and 4.5).

The final grade of the academic record is obtained according to the final grading table:

Programming (Block A)				
		Note $\geq 5$	Note $< 5$	<pdire="ltr">NA
Computation (Block B)	Note $\geq 5$	60% block A + 40% block B	Min (60% block A + 40% block B, 4.5)	Min (40% block B, 3 , 4)
	Note $< 5$	Min (60% block A + 40% block B, 4.5)	Min (60% block A + 40% block B, 4.5)	Min (40% block B, 3,4)
	NA	Min(60% block A, 3,4)	Min (60% block A, 3,4)	NA

Not Assessable (NA): Any student who delivers a practice or scheduled activity will receive a grade. It will only be considered non-evaluable, overall, if any evaluable activity is delivered. In case of the student delivers a practice or scheduled activity in one of the blocks but not in the other, it will be considered that the block that has not presented anything will have a 0.

Honors matriculations: Awarding an honors matriculation qualification is the decision of the teaching staff responsible for the subject. UAB regulations indicate that MH can only be granted to students who have obtained a final grade equal to or higher than 9.00 and their number cannot exceed 5% of the total number of enrolled students.

In the event that the number of students with a grade greater than or equal to 9 is more than 5% of the total enrolled, the criteria to be applied in the order listed below will be the following:

1. Priority is given to those who have not gone to any recovery.
2. Priority is given to those with grades  $\geq 9$  in both blocks.
3. Priority is given to those with grades  $\geq 9$  in all parts of the 2 blocks.



4. They are prioritized in order of average grade.

#### Irregularities by the student, copying ...

Without prejudice to other disciplinary measures that are 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 result in failing that assessment activity with a zero (0). Assessment activities qualified in this way and for this reason 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, with no opportunity to recover it in the same course.

#### Evaluation of repeating students

Anyone who repeats the subject and has one of the two blocks fully approved (Theory (T) and Practices (PL) of the block, with grades  $\geq 5$ ) will not need to re-evaluate this block the following year.

For block A, if the subject "Programming methodology" was passed in the previous year and the review exercises proposed in that subject were approved, a summary assessment of the block can be made, without having to do all the evaluation activities of the block. In this case, the mark of the block will be 5.

To be able to opt for this differentiated assessment, the interested person must expressly request it from the teaching staff by email to: for the overcome Programming block (xavier.roca@uab.cat or robert.benavente@uab.cat) for the overcome Computing block (miquelangel.senar@uab.cat) no later than 15 days after the start of classes in each Block.

It is the student's responsibility to check that the responsible teaching staff correctly publishes the grade obtained in the course that is ending. Under no circumstances are notes kept of any of the evaluation activities of a block that has been suspended in a previous course, as could be the case of having suspended the theory and having passed the practices of a block.

Note: The proposed assessment may undergo some modification depending on any restrictions on attendance imposed by the health authorities.

## **Bibliography**

- J. Pujol: Algorismes i Programes. Servei de publicacions de la UAB, 1996.
- E. Valveny, R. Benavente, A. Lapedriza, M. Ferrer, J. García: Programació en Llenguatge C. Amb 56 problemes resolts i comentats. Servei de publicacions UAB, 2009.
- L. Joyanes Aguilar: Fundamentos de Programación: Algoritmos, Estructuras de Datos y Objetos. 5ª Edición, Mc. Graw-Hill, 2008.
- L. Joyanes Aguilar, L. Rodríguez Baena, M. Fernández Azuela: Fundamentos de Programación. Libro de Problemas. 2ª Edición, Mc Graw-Hill, 2003.
- B.W. Kernighan, D.M. Ritchie: El lenguaje de programación C. 2ª Edición, Prentice Hall, 1986.
- L. Joyanes, A. Castillo, L. Sánchez, I. Zahonero: Programación en C: libro de problemas, Mc Graw-Hill, 2005.
- L. Joyanes, I. Zahonero: Programación en C: metodología, estructura de datos y objetos, Mc Graw-Hill, 2001.
- B.W. Kernighan, R. Pike: La Práctica de la Programación. Pearson Educación, 2000.
- A. Prieto, A. Lloris, J.C. Torres: Introducción a la Informática. Mc Graw-Hill, 4ª Edición, 2006.
- A. Prieto, B. Prieto: Conceptos de Informática. Mc Graw-Hill, Schaum, 2006.

## **Software**

### Programming (BLOCK A) & Computation (BLOCK B)

## Language list

Name	Group	Language	Semester	Turn
(PAUL) Classroom practices	411	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	412	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	431	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	432	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	451	Catalan	first semester	afternoon
(PAUL) Classroom practices	452	Catalan	first semester	afternoon
(PAUL) Classroom practices	471	Catalan	first semester	afternoon
(PLAB) Practical laboratories	411	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	412	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	413	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	414	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	415	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	416	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	417	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	418	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	419	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	420	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	421	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	422	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	423	Catalan/Spanish	first semester	afternoon
(TE) Theory	41	Catalan	first semester	morning-mixed
(TE) Theory	43	Catalan	first semester	morning-mixed
(TE) Theory	45	Catalan/Spanish	first semester	afternoon