Universitat Autònoma de Barcelona

2020/2021

## Fundamentals of Computing

Code: 103806
ECTS Credits: 9

| Degree | Type | Year | Semester |
|---|---|---|---|
| 2502441 Computer 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 Sorribes Gomis

Email: Joan.Sorribes@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

Francesc Xavier Roca Marva

Joan Sorribes Gomis

Katerine Diaz Chito

Lidia Garrido Sanz

Robert Benavente Vidal

Joan Josep Piedrafita Farras

## External teachers

xavier cano

## 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 unitsof 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 discreet 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 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.

Unit 6: Problem solving. Complexes: Modular Design

Phases in the development of programs. Architectural design Introduction to the modular design methodology. Descending design

BLOCK 3: COMPUTING INTRODUCTION

Unit 7: 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 8: 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 9: Machine language and assembler

Direction. Elementary didactic computer Assembly language

Unit 10: Peripherals.

Unit 11: Operating system. Evolution. Management: processor, memory, I / O, files

Unit 12: Concepts on computer networks

Data transmission. Computer networks. Internet

## Methodology

The subject has twoTeaching will be offered on campus or in an on-campus and remote hybrid format depending on the number of students per group and the size of the rooms at 50% capacity. 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 theplatform 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
- To 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 theplatform 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 willbe 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 deliveredafter 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 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.

The proposed teaching methodology may undergo some modifications according to the restrictions imposed by the health authorities on on-campus courses

## Activities

| 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 |

## Assessment

Programming 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 | Consult Planning | Consultation Planning | 50 % | T> = 5 | 30% |
| Self-assessment questionnaire of achievement concepts | Weekly and Until theory recovery exam | No | 5% | Not.<br>Serves to unlock access to problems | 3% |
| Problems resolution (P) Individual | Up to theory recovery exam<br>It can only be accessed if the achievement questionnaire has been previously completed<br>(some ones will have deadline) | No | 15% | No | 9% |
| Laboratory practice (PL) Group | Consult Planning<br>Compose of a test (PLt) and the correction ofthe final delivery (PLf) | Consult Planning | 30% | PL = 30% PLt + 70% PLf<br>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 | Consult Planning | Consultation Planning | 60% | T> = 5 | 24% |
| Resolution Problems (P) Individual | Check Plan | No | 10% | No | 4% |
| Laboratory Practicum (PL) Grupa I | Consult Planning<br>You have three deliveries, one for each full practice.<br>P1, P2 & P3 | Consult Planning | 30% | PL= 33%P1 + 33% P2 + 33% P3<br>P1,P2,P3 >= 3.5 i PL>=5 | 12% |

The note of each block is the weighted sum of the assessment activities with their corresponding percentage. Theresult of the weighted sum must be> = 5 to consider the block approved.

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

The evaluation and delivery dates will be published in the document manager chosen by each Blog (Caronte (Block A) or Virtual Campus (Block B) and may be subject to changes in the programming for reasons of adaptation to possible incidents. 'will inform the document manager about these changes since it is understood that this is the usual platform for exchanging information between teachers and students.

Procedure for the review of qualifications

The student will have the right to review the theoretical examinations (T). The site, date and time of review will be published on the day the notes are published. If the student does not submit to this review, they will not be reviewed later. Only in justified cases can a posteriori review of the established date and always up to a maximum of 7 calendar days afterwards.

Of the notes of the laboratory practice (PL) there will be no review because the assessment is carried out in front of the students.

Qualifications

Not Evaluable (NA): Any student who submits a practice or a scheduled evaluation will note. It will only be considered non-evaluable if you do not deliver any evaluable activity. In case the student submits a practice or assessment scheduled in one of the blogs and not in the other, it will be considered that the blog that has not presented anything will have a 0.

Final Note: The two blogs will be evaluated independently. To obtain the note of the file, the two parts must be approved separately (Note> = 5).

The note of each block comes from the weighted sum according to the criteria set out in the section of evaluation activities. If the theory (T) or the practices (PL) have a note inferior to 5 the note will come out of the following formula:

<palign = "center"> Minimum (apply weighted sum i 4.9)

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

| | | Programming (Block A) | | |
|---|---|---|---|---|
| | | Note> = 5 | Note <5 | NA |
| Computation (Block B) | Note> = 5 | 60% block A + 40% block B | Min (60% block A + 40% block B, 4.9) | Min (40% block B, 3 , 4) |

|  |  |  |  |  |
|---|---|---|---|---|
|  | Note <5 | Min (60% block A + 40% block B, 4.9) | Min (60% block A + 40% block B, 4.9) | Min (40% block B, 3,4) |
|  | NA | Min (60% block A, 3,4) | Min (60% block A, 3,4) | NA |

Honor: matrorsGranting an honorific matriculation qualification is the decision of the faculty responsible for the subject. The regulations of the UAB indicatethat 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. In the case that the number of studentswitha mark greater than or equal to 9 ismore than 5% of the total of enrolled, the criteria to apply, in the order listedbelow, will be the following:

1. Priority is given to those who have not gone to any recovery exam
2. Priorityis given to those with note> 9 to both blocks.
3. Priority is given to those who have note> 9 on all parts of the 2 blocks.
4. They are prioritized by average note order.

Irregularities by the student, copy and plagiarism

Notwithstanding other disciplinary measures that are deemed appropriate, and in accordance with current academic regulations, irregularities committed by a student who may lead to a variation of the qualification will be qualified with 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 the opportunity to recover it in the same course.

Evaluation of repeating students

Who repeats the subject and has one of the two blocks approved completely (Theory (T) and Practices (PL) with notes> = 5) will not have to be evaluated again of this block.

In order to be able to opt for this differentiated assessment, the interested person must ask the teaching staff by e-mail Programming (xavier.roca@uab.cat or robert.benavente@uab.cat) Computing (joan.sorribes@uab.cat) as a no later than 15 days after the start of classes.

It is the responsibility of the person interested to verify that the responsible teaching staff correctly publishes the grade obtained last academic year. In no case are notes of any of the evaluation activities of a block that has been suspended in a previous course, such as thecase may have suspended the theory and have passed the practices.

The proposed evaluation activities may undergo some changes according to the restrictions imposed by the health authorities on on-campus courses

## 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 |

## 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. 3ª Edición, Mc. Graw-Hill, 2003.
- *L. Joyanes Aguilar, L. Rodriguez 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, 2002.
- *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, 2005.
- *A. Prieto, B. Prieto*: Conceptos de Informática. Mc Graw-Hill, Schaum, 2006.